Chapter 1

Gradient Based Policy Optimization of Constrained Unichain Markov Decision Processes

Vikram Krishnamurthy* and Felisa J. Vázquez Abad†

We present stochastic approximation algorithms for computing the locally optimal policy of a constrained, average cost, finite state Markov Decision Process. The stochastic approximation algorithms require estimation of the gradient of the cost function with respect to the parameter that characterizes the randomized policy. We propose a spherical coordinate parameterization and implement a simulation based gradient estimation scheme using using potential realization factors. In this paper we analyse boundedness of moments of the estimators and provide a new analysis for the computational requirements of the estimators. We present a stochastic version of a primal dual (augmented) Lagrange multiplier method for the constrained algorithm. We show that the "large sample size" limit of the procedure is unbiased, but for small sample size (which is relevant for on-line real time operation) it is biased. We give an explicit expression of the asymptotic bias and discuss several possibilities for bias reduction. Numerical examples are given to illustrate the performance of the algorithms. In particular we present a case study in transmission control in wireless communications where the optimal policy has a simplified threshold structure and apply our stochastic approximation method to this example.

1. Introduction

Let S denote an arbitrary finite set called the *state space*. Let $\mathcal{U}_i$, $i \in S$ denote an arbitrary collection of finite sets called *action sets*. A Markov Decision Process [24] (MDP) $\{X_n\}$ with finite state space S evolves as follows. When the system is in state $i \in S$, a finite number of possible actions from the finite set $\mathcal{U}_i$ can be taken. Let u_n denote the action taken by the decision maker at time n and let $d(i)+1$ denote the cardinality of the action set $\mathcal{U}_i$. The evolution of the system is Markovian with a transition probability matrix $A(u)$ that depends on the action

*Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, V6T 1Z4, Canada. vikramk@ece.ubc.ca
†Department of Computer Science, Hunter College - City University of New York, NY 10065. felisav@hunter.cuny.edu

$u \in \mathcal{U}_i$, that is for $i, j \in S$,

$$A_{ij}(u) \stackrel{\triangle}{=} \mathbb{P}[X_{n+1} = j | X_n = i, u_n = u], \quad u \in \mathcal{U}_i, \quad n = 0, 1, \ldots \tag{1}$$

Denote by $\mathfrak{F}_n, n \geq 1$ the σ-algebra generated by the observed *system trajectory* $(X_0, \ldots, X_n, u_0, \ldots, u_{n-1})$ and set $\mathfrak{F}_0$ as the σ-algebra generated by X_0. The filtration of the process is the increasing sequence of σ-algebras $\{\mathfrak{F}_n, n \geq 0\}$. Define the set of **admissible** policies $\mathcal{D} = \{\mathbf{u} = \{u_n\} : u_n \text{ is measurable w.r.t. } \mathfrak{F}_n, \forall n \in \mathbb{N}\}$. To avoid being distracted by technicalities, except for Sec.6, we assume that all elements of $A(u)$ for each u are non-zero so that the resulting process is ergodic. In Sec.6, a wireless telecommunications example is presented and sufficient conditions are given so that every feasibly policy induces a recurrent Markov chain.

The cost incurred at stage n is a known bounded function $c(X_n, u_n) \geq 0$ where $c : S \times \mathcal{U} \to \mathbb{R}$. For any admissible policy $\mathbf{u} \in \mathcal{D}$, let $\mathbb{E}_{\mathbf{u}}$ denote the corresponding expectation and define the infinite horizon average cost

$$J_{x_0}(\mathbf{u}) = \lim_{N\to\infty} \sup \frac{1}{N} \mathbb{E}_{\mathbf{u}} \left[\sum_{n=1}^{N} c(X_n, u_n) \mid X_0 = x_0 \right]. \tag{2}$$

Motivated by several problems in telecommunication network optimization such as admission control in wireless networks [33; 22; 11], we consider the cost (2), subject to L sample path constraints

$$\mathbb{P}_{\mathbf{u}} \left[\lim \sup_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \beta_l(X_n, u_n) \leq \gamma_l \right] = 1, \quad l = 1, 2, \ldots, L,$$

where $\beta_l : S \times \mathcal{U} \to \mathbb{R}$ are known bounded functions and γ_l are known constants. These are used, for example, in admission control of telecommunication networks to depict quality of service (QoS) constraints, see [33; 22]. For ergodic MDPs, these sample path constraints are equivalent to the average constraints [25]:

$$\lim_{N\to\infty} \sup \frac{1}{N} \mathbb{E}_{\mathbf{u}} \left[\sum_{n=1}^{N} \beta_l(X_n, u_n) \right] \leq \gamma_l, \quad l = 1, \ldots, L. \tag{3}$$

The aim is to compute an optimal policy $\mathbf{u}^* \in \mathcal{D}$ that satisfies

$$J_{x_0}(\mathbf{u}^*) = \inf_{\mathbf{u}\in\mathcal{D}} J_{x_0}(\mathbf{u}) \quad \forall x_0 \in S, \tag{4}$$

that is, $\mathbf{u}^*$ has the minimum cost for all initial states $x_0 \in S$ subject to the constraints (3). It is well known [2] that if $L > 0$ then the optimal policy $\mathbf{u}^*$ is *randomized* for at most L of the states. If the transition probabilities $A_{ij}(u)$ (1) are known, then the optimal policy $\mathbf{u}^*$ for the constrained MDP (2), (3) is straightforwardly computed as the solution of a linear programming problem.

Objectives. Problem (2), (3) is an adaptive constrained Markov decision process problem. This paper presents stochastic approximation algorithms for adaptively computing the optimal policy $\mathbf{u}^*$ of the above constrained MDP (2), (3).

There are two methodologies that are used in the literature for solving stochastic adaptive control problems: ***direct methods***, where the unknown transition probabilities $A_{ij}(u)$ are estimated *simultaneously* while updating the control policy, and ***implicit methods*** – such as simulation based methods, where the transition probabilities are not directly estimated in order to compute the control policy. In this paper, we focus on implicit simulation-based algorithms for solving the MDP (2), (3). By simulation based we mean that although the transition probabilities $A(u)$, $u \in \mathcal{U}_i$, $i \in S$ are unknown, the decision maker can observe the system trajectory under any choice of control actions $\mathbf{u} = \{u_n\}$. In other words, the adaptive control algorithms we present are adapted to the filtration $\{\mathfrak{F}_n; n \geq 0\}$ defined above. Moreover, these algorithms can deal with slowly time varying transition probabilities $A(u)$.

Neurodynamic programming methods [7] such as Q-learning and temporal difference methods are also examples of simulation based implicit methods that have been widely used to solve unconstrained MDPs where the optimal policy $\mathbf{u}^*$ is a pure policy, that is, u_n^* is a deterministic function of X_n. However, for the constrained MDP (2), (3), since the optimal policy is randomized, there seems to be no obvious way of modifying such methods.

Summary of Main results In this paper we present a stochastic version of gradient-based optimisation methods. In Section 4, we derive a consistent weak derivative (WD) "phantom" estimator. This gradient estimator does not require explicit knowledge of the transition probabilities $A(u)$ – so for brevity we call it a "parameter free" gradient estimator.

Section 4.2 presents the implementation of our WD estimators over short batch lengths to facilitate real time control of the MDP. The key advantage is that they are amenable to adaptive control of MDPs with time varying parameters. Moreover, these gradient estimators yield several orders of magnitude reduction in variance compared to the score function method in [4; 3]. However, as we discuss in Section 4.2, a non negligible asymptotic bias may occur. We derive the statistical properties (consistency, bias for small batch size, and efficiency) of the gradient estimates. We give stochastic bounds for the number of parallel phantoms, which allows us to approximate the moments of the coupling time as well as the computational complexity of the algorithm. To our knowledge, these results, which are applicable to realisation perturbation factors, are new.

Putting the parameter free gradient estimation algorithm into a stochastic gradient algorithm results in a new simulation based algorithm for the adaptive control of a constrained MDP. The control problem (4) under constrains (3) can be reformulated as an optimization problem using a parametrization, as will be stated in Section 2. The main challenge that we address in this paper is as follows: Given N observations of the process under a given policy parametrized by the control parameter $\alpha(n)$, say $Z_{(n-1)N}, \ldots, Z_{nN}$, and without explicit knowledge of the transition probabilities $P_{ij}(u)$, adjust the control variable $\alpha(n+1)$ to approximate the optimal control. We assume that the actual instantaneous cost and constraints values are observed exactly. In the stationary case where the transition probabilities, cost and constraint functions do not change, one usually constructs algorithms such that $\|\alpha(n)-\alpha*\|$ is "small" in some sense, usually with probability one. For tracking however, we focus on an algorithm with adaptive capabilities that will get close to the current optimum, while at the same time, it reacts to changes in the environment. There are two important results that we present in this context.

First, because the constraints in the MDP are in the form of long term averages that are not known to the decision maker, it is not possible to use stochastic approximation algorithms with the usual gradient projection methods in [18], for example. We present primal dual based stochastic approximation algorithms with a penalty function and augmented Lagrangian (multiplier) algorithms are presented for solving the time varying constrained MDP. Weak convergence of the action probability estimates to the optimal action probabilties is established when the sample size for the estimation is large. However, for the short batch length implementation, we show that the algorithm is asymptotically suboptimal and we characterise the asymptotic bias Sec.5.3 illustrates the performance of the algorithms on a constrained MDP with time varying transition probabilities.

Second, because the action probabilities must always add up to one and be non negative, there is strong motivation to develop a parametrization that ensures feasibility of the estimates generated by the stochastic approximation algorithm at every step. We parameterize the action probabilities of the constrained MDP using *spherical coordinates*. This parametrization is particularly well suited to the MDP problem and has superior convergence rate, see discussion in Sec.2.4. In [4; 3], a different parameterization is used which is similar to the generalized gradient approach of [30].

Sec.6, gives a case study of how the results in the paper can be used to estimate a monotone scheduling policy that arises in transmission scheduling over fading wireless communication channels. We prove that the optimal scheduling policy is a randomized mixture of two monotone (threshold) policies – this proof is of independent interest as it generalizes the well known results of the classical paper

[10] (see also [26]) to the case of a correlated Markov chain channel and with delay constraints.

2. Problem Formulation and Discussion

In this section we formulate the constrained MDP (2), (3) as a stochastic optimization problem in terms of parameterized action probabilities. Then for convenience a summary of the key algorithms in this paper is given. Finally we briefly summarize how our approach differs from two other approaches in the literature.

2.1. *Spherically Parameterized Randomized Policies*

The randomized optimal policy for the above constrained MDP can be defined in terms of the action probabilities θ parameterized by ψ as:

$$\mathbb{P}[u_n = a|X_n = i] = \theta_{ia}(\psi), \quad a \in \mathcal{U}_i,\ i \in S \tag{5}$$
$$\text{where } \theta_{ia}(\psi) \geq 0, \quad \sum_{a \in \mathcal{U}_i} \theta_{ia}(\psi) = 1 \text{ for every state } i \in S.$$

Here $\psi \in \Psi$ is a finite dimensional vector which parameterizes the action probabilities θ, and Ψ is some suitably defined compact subset of the Euclidean space. The unconstrained problem with pure optimal policy is a degenerate case, where for each $i \in S$, $\theta_{ia} = 1$ for some $a \in \mathcal{U}_i$.

The most obvious parameterization ψ for θ – which we will call *canonical coordinates* is to choose $\psi = \theta$. Thus $\psi = \{\psi_i\} = \{\theta_i\}$, $i \in S$ is the set of action probability vectors $(\psi_{ia}; a \in \mathcal{U}_i)$ satisfying (5). As discussed in Sec.2.4, this parameterization has several disadvantages.

In this paper we use a more convenient *spherical coordinate* parameterization $\psi = \alpha$ that automatically ensures the feasibility of $\theta(\alpha)$ (that is, the constraints in (5) hold) without imposing a hard constraint on α. Adapted to our MDP problem, it reads as follows: Fix the control agent $i \in S$. Suppose without loss of generality that $\mathcal{U}_i = \{0, \ldots, d(i)\}$. To each value $\theta_{ia}, a \in \mathcal{U}_i$ associate the values $\lambda_{ia} = \sqrt{\theta_{ia}}$. Then (5) yields $\sum_{a \in \mathcal{U}_i} \lambda_{ia}^2 = 1$, and λ_{ia} can be interpreted as the coordinates of a vector that lies on the surface of the unit sphere in $\mathbb{R}^{d(i)+1}$, where $d(i)+1$ is the size of $\mathcal{U}_i$ (that is, number of actions). In spherical coordinates, the angles are $\alpha_{ip}, p = 1, \ldots d(i)$, and the radius is always of size unity. For $d(i) \geq 1$,

the spherical coordinates parameterization α satisfies:

$$\theta_{ia}(\alpha) = \lambda_{ia}^2, \quad \lambda_{ia} = \begin{cases} \cos(\alpha_{i,1}) & \text{if } a = 0 \\ \cos(\alpha_{i,(a+1)}) \prod_{k=1}^{a} \sin(\alpha_{i,k}) & 1 \le a \le d(i) - 1 \\ \sin(\alpha_{i,d(i))}) \prod_{k=1}^{d(i)-1} \sin(\alpha_{i,k}) & a = d(i) \end{cases}. \tag{6}$$

Note that $\theta_{ia}(\alpha) = \lambda_{ia}^2$ is an analytic function of α, that is, infinitely differentiable in α. It is clear that under this α parameterization, the control variables $\alpha_{ip}, p \in \{1, \ldots, d(i)\}$ do not need to satisfy any constraints in order for $\theta(\alpha)$ to be feasible. Furthermore, since $\theta_{ia}(\alpha) = \lambda_{ia}^2$ involves even powers of $\sin(\alpha_{i,p})$ and $\cos(\alpha_{i,p})$, it suffices to consider $\alpha_{ip} \in \boldsymbol{\alpha}$ where $\boldsymbol{\alpha}$ denotes the compact set

$$\boldsymbol{\alpha} = \{\alpha_{ip} \in [0, \pi/2];\ i \in S,\ p \in \{1, \ldots, d(i)\}\}, \tag{7}$$

that is, for any $\alpha \in \mathbb{R}^{d(i)}$, there is a unique $\alpha \in \boldsymbol{\alpha}$ which yields the same value of $\theta(\alpha)$. Let $\boldsymbol{\alpha}^o$ denote the interior of the set $\boldsymbol{\alpha}$. Finally, define the compact set $\boldsymbol{\alpha}^\mu \subset \boldsymbol{\alpha}^o$ for user defined small parameter $\mu > 0$ as

$$\boldsymbol{\alpha}^\mu = \{\alpha_{ip} \in [\mu, \pi/2 - \mu];\ i \in S,\ p \in \{1, \ldots, d(i)\}\}. \tag{8}$$

Note that $\boldsymbol{\alpha}^\mu$ is the set obtained by removing small intervals at 0 and $\pi/2$ from $\boldsymbol{\alpha}$. As discussed in detail after the statement of Proposition 1.1 in Sec.5, excluding 0 and $\pi/2$ is necessary to prove convergence of the algorithms we propose – however, since μ can be chosen arbitrarily small, it is not important in the actual algorithmic implementation.

2.2. *Parameterized Constrained MDP Formulation*

We now formulate the above MDP problem as a stochastic optimization problem where the instantaneous random cost is independent of α but the expectation is with respect to a measure parameterized by α. Such a "parameterized integrator" formulation is common in gradient estimation, see [23], and will be subsequently used to derive our gradient estimators. Consider the augmented (homogeneous) Markov chain $Z_n \stackrel{\triangle}{=} (X_n, u_n)$ with state space $\mathcal{Z} = S \times \mathcal{U}$ and transition probabilities parameterized by α given by

$$P_{i,a,j,a'}(\alpha) \stackrel{\triangle}{=} \mathbb{P}(X_{n+1} = j, u_{n+1} = a' \mid X_n = i, u_n = a) = \theta_{j,a'}(\alpha)\, A_{ij}(a), \quad i, j \in S, a \in \mathcal{U}_i, a' \in \mathcal{U}_j \tag{9}$$

It follows that for any $\alpha \in \boldsymbol{\alpha}^o$ (interior of set $\boldsymbol{\alpha}$), the chain $\{Z_n\}$ is ergodic, and it possesses a unique invariant probability measure $\pi_{i,a}(\alpha); i \in S,\ a \in \mathcal{U}_i$. Let

$\mathbb{E}_{\pi(\alpha)}$ denote expectation w.r.t measure $\pi(\alpha)$ parameterized by α. From (2) we have $J_{x_0}(\mathbf{u}) = C(\theta(\alpha))$, where

$$C(\theta(\alpha)) \stackrel{\triangle}{=} \mathbb{E}_{\pi(\alpha)}[c(Z)] = \sum_{i \in S} \sum_{a \in \mathcal{U}_i} \pi_{i,a}(\alpha) c(i,a). \tag{10}$$

We subsequently denote $C(\theta(\alpha))$ as $C(\alpha)$, whenever it is convenient. The L constraints (3) can be expressed as

$$B_l(\alpha) \stackrel{\triangle}{=} \mathbb{E}_{\pi(\alpha)}[\beta_l(Z)] - \gamma_l = \sum_{i \in S} \sum_{a \in \mathcal{U}_i} \pi_{i,a}(\alpha) \beta_l(i,a) - \gamma_l \leq 0, \quad l = 1, \ldots, L. \tag{11}$$

Define $B = (B_1, \ldots, B_l)$. Thus the optimization problem (4) with constraints (3) can be written as

$$\textbf{Problem S1:} \min_{\alpha \in \boldsymbol{\alpha}^\mu} C(\alpha) \tag{12}$$

$$\text{subject to:} \quad B_l(\alpha) \leq 0, \quad l = 1, \ldots, L. \tag{13}$$

The constraints $B_l(\alpha)$ in (13) will be called **MDP constraints**. As should be evident from this formulation, the optimal control problem (12), (13) depends uniquely on the *invariant* distribution $\pi(\alpha)$ of the chain, rather than the (unknown) transition probabilities $A_{ij}(u)$.

It is important to note that in this paper, the expected cost $C(\alpha)$ and expected constraints $B_l(\alpha) \leq 0$ are not known to the decision maker (since the transition probabilities are not known). Our aim is to devise a recursive (online) stochastic approximation algorithm to solve Problem S1 without explicit knowledge of the transition probabilities $A(a)$. Such an algorithm operates recursively on the observed system trajectory $(X_0, \ldots, X_n, u_0, \ldots, u_{n-1})$ to yield a sequence of estimates $\{\alpha(n)\}$ of the optimal solution. If the unknown dynamics $A(a)$ of the MDP are constant, then the proposed algorithm ensures that the estimates $\alpha(n)$ approach the optimal solution. On the other hand, if the unknown underlying dynamics $A(a)$ evolves slowly with time or jump changes infrequently, then the proposed algorithm will track the optimal trajectory in a sense to be made clear later. Stochastic approximation algorithms for tracking slowly evolving parameters has been widely studied in the literature, see [5; 28]. Tracking parameters that jump changes infrequently has been analysed more recently in [32].

Assumption 1.1. The minima α^* of Problem S1 (that is, (12), (13)) are regular, that is, $\nabla_\alpha B_l(\alpha^*)$, $l = 1, \ldots, L$ are linearly independent. Also, α^* belongs to the

set of Kuhn Tucker points

$$\text{KT} = \left\{ \alpha^* \in \boldsymbol{\alpha}^\mu : \exists\, \mu_l \geq 0,\, l = 1, \ldots, L \text{ such that } \quad \nabla_\alpha C + \nabla_\alpha B \mu = 0, \quad B'\mu = 0 \right\} \tag{14}$$

where $\mu = (\mu_1 \ldots, \mu_L)'$. Moreover α^* satisfies the second order sufficiency condition $\nabla_\alpha{}^2 C(\alpha^*) + \nabla_\alpha{}^2 B(\alpha^*)\mu > 0$ (positive definite) on the subspace $\{y \in \mathbb{R}^L : \nabla_\alpha B_l(\alpha^*) y = 0\}$ for all $l : B_l(\alpha^*) = 0, \mu_l > 0$.

2.3. *User's Guide*

A summary of the key equations for implementing the learning based algorithm proposed in this paper in spherical coordinates for constrained MDP Problem S1 ((12), (13)) is as follows:

Input Parameters: Cost matrix $(c(i,a))$, constraint matrix $(\beta(i,a))$, batch size N.

Step 0. Initialize: Set $n = 0$, initialize $\alpha(n) \in \boldsymbol{\alpha}^o$ and vector $\lambda(n) \in \mathbb{R}^L_+$. [‡]

Step 1. System Trajectory Observation: Observe MDP over batch $I_n \stackrel{\triangle}{=} \{k \in [nN, (n+1)N-1]\}$ using randomized policy $\theta(\alpha(n))$ of (6) and compute estimate $\hat{B}(n)$ of the constraints, (cf. (50) or (52) of Sec.5.3).

Step 2. Gradient Estimation without explicit knowledge of parameters: Compute gradient estimates $\widehat{\nabla_\alpha C}(n)$, $\widehat{\nabla_\alpha B}(n)$ over the batch I_n using the WD phantom estimates $\widehat{\widehat{G}}_n$ given in (41).

Step 3. Update Policy $\theta(\alpha(n))$ using constrained stochastic gradient algorithm: Use a penalty function primal dual based stochastic approximation algorithm to update α as follows:

$$\alpha(n+1) = \alpha(n) - \epsilon \left(\widehat{\nabla_\alpha C}(n) + \widehat{\nabla_\alpha B}(n) \max\left[0, \lambda(n) + \rho \widehat{B}(n)\right] \right) \tag{15}$$

$$\lambda(n+1) = \max\left[\left(1 - \frac{\epsilon}{\rho}\right) \lambda(n), \lambda(n) + \epsilon \widehat{B}(n) \right]. \tag{16}$$

The "penalization" ρ is a suitably large positive constant and $\max[\cdot\cdot]$ above is taken element wise, see (22), (23).

Step 4. Set $n = n + 1$ and go to Step 1.

Remark 1.1. 1. For large batch size N, the bias of the estimates $\hat{B}$ and $\widehat{\nabla_\alpha B}$ are $O(1/N)$ and the algorithm (15), (16) is asymptotically optimal. However, for fast

[‡]More precisely, $\alpha(0)$ needs to be initialized in $\boldsymbol{\alpha}^\mu$, where $\boldsymbol{\alpha}^\mu \subset \boldsymbol{\alpha}^o$ excludes a small μ size ball around the boundary of $\boldsymbol{\alpha}$, see (8) and Sec.5.1.

tracking of time-varying MDPs it is necessary to choose N small. In this case, the main source of bias in the estimation of α^* using (15), (16) is the covariance of $\widehat{\nabla_\alpha B}$ with $\hat{B}$. Several approaches to deal with this bias are discussed in Sec.5.3.
2. Another alternative to (16) is to update λ via a multiplier (augmented Lagrangian) algorithm (see Sec.3.2). A third alternative is to fix λ. For sufficiently large ρ, $\alpha(n)$ will converge to $\alpha(\infty)$ which is in a pre-specified ball around a local minimum α^*.
3. If the true parameters of the MDP jump change at infrequent intervals, then iterate averaging [19] (as long as the minimal window of averaging is smaller than the jump change time) and adaptive step size algorithms can be implemented in the above stochastic approximation algorithms to improve efficiency and tracking capabilities.
4. For the special case of two actions and a monotone policy (see Section 6), then $\alpha(n)$ can be further parameterized and the dimension of the optimization Problem S1 can be substantially reduced.

2.4. *Discussion of Other approaches in Literature*

Before presenting the details of the algorithm proposed in this paper, we briefly summarize two works in the literature that also use stochastic approximation methods to solve MDPs. It is well known [24] that Problem S1 formulated in terms of the invariant measure π is the following linear program:

$$\min_{\pi} \sum_{i\in S}\sum_{a\in\mathcal{U}_i} \pi_{ia} c(i,a) \tag{17}$$

$$\text{subject to } \sum_{i\in S}\sum_{a\in\mathcal{U}_i} \pi_{ia}\beta_l(i,a) < \gamma_l$$

$$\sum_{a\in\mathcal{U}_j} \pi_{ja} = \sum_{i\in S}\sum_{a\in\mathcal{U}_j} \pi_{ia} A_{ij}(a), \qquad \sum_{i\in S, a\in\mathcal{U}_j} \pi_{ia} = 1, \quad 0 \le \pi_{ia} \le 1, \quad i\in S, a\in\mathcal{U}_j.$$

With θ^* and π^* denoting the optimal solutions of (12) and (17), respectively, it is straightforward to show that

$$\theta^*_{ia} = \frac{\pi^*_{ia}}{\sum_{u\in\mathcal{U}_i}\pi^*_{iu}}. \tag{18}$$

The closest approach to our paper is that presented in [4; 3]. The MDP in [4; 3] is without constraints and uses the parameterization

$$\theta_{ia}(\psi) = \frac{e^{\psi_{ia}}}{\sum_{u\in\mathcal{U}_i} e^{\psi_{iu}}}, \quad \psi_{ia}\in\mathbb{R}, i\in S,\ a\in\mathcal{U}_i.$$

This exponential parameterization satisfies

$$\frac{\partial \theta_{iu}}{\partial \psi_{ia}} = \begin{cases} \theta_{iu}(1-\theta_{iu}) & u = a \\ -\theta_{iu}\theta_{ia} & u \neq a. \end{cases}$$

Using the chain rule of differentiation on the cost function $C(\theta(\psi))$ (defined similarly to (10))

$$\frac{\partial}{\partial \psi_{ia}} C[\theta(\psi)] = \sum_{u \in \mathcal{U}_i} \frac{\partial}{\partial \theta_{ia}} C(\theta) \left(\frac{\partial \theta_{iu}}{\partial \psi_{ia}} \right) = \theta_{ia} \left(\frac{\partial}{\partial \theta_{ia}} C(\theta) - \sum_{u \in \mathcal{U}_i} \theta_{iu} \frac{\partial}{\partial \theta_{ia}} C(\theta) \right), \tag{19}$$

which is identical to the *Generalized Gradient* of [30]. In our report [1], we explain why this formulation yields the appropriate descent directional derivative for canonical coordinates of in Sec.2.1. See also [30] and references therein. However, gradient algorithms based on this parameterization can exhibit slow convergence, particularly when the optimal probability vector θ^* is degenerate. When a component of θ_{ia} is zero, (19) is zero and hence a gradient based algorithm remains at this point. Because the drift of the update is proportional to the size of the updated component, as a component approaches zero, the magnitude of future updates decreases (to prevent crossing outside the feasible set). This mechanism slows down convergence of the gradient algorithm using canonical coordinates $\psi = \theta$, particularly close to the optimal solution if this has components representing pure strategies, as is often the case. We refer the reader to [1] for numerical examples that demonstrate that the parameterization involving spherical coordinates has superior convergence properties compared to canonical coordinates.

In addition, the approach for derivative estimation in [4; 3] is via the Score Function method, which usually suffers from unbounded variance for infinite horizon costs. To alleviate this problem, the authors use a forgetting factor that introduces a bias in the derivative estimation. Our derivative estimators are more efficient and consistent, with provably bounded variance over infinite horizon, in $\boldsymbol{\alpha}^o$. In Sec.7.1, numerical examples show that the variance of the score function method is several orders of magnitude larger than that of the measured valued derivative estimator.

The above methods all use a "simulation optimization" approach, where almost sure convergence to the true optimal value can be shown under an appropriate choice of parameters of the algorithms. In particular, all stochastic approximations involved in the above mentioned methodologies use decreasing step size. One of the motivations of the present work is to implement a stochastic approximation procedure with constant step size in order for the controlled Markov chain

to be able to deal with tracking slowly varying external conditions, which result in slowly varying $A(u)$.

Remark 1.2. Our MDP setting assumes perfect observation of the process $\{X_n\}$. The paper [4; 3] considers a partially observed MDP (POMDP), but assumes that the observations Y_n of the process X_n belong to a finite set. In that work they consider suboptimal strategies of the form $\theta_{ia} = \mathbb{P}\{u_n = a \,|\, Y_n = i\}$. Such a policy is clearly not optimal for a POMDP since the optimal policy is a measurable function of the history $(Y_1, \ldots, Y_k, u_1, \ldots, u_{k-1})$, which is summarized by a continuous-valued information state. Such suboptimal POMDP models are a special case of the problem considered here and our method can be applied in a straightforward manner. We refer the reader to [14; 16; 15] for structural results in POMDPs.

3. Ordinary Differential Equations for Solving Constrained MDPs

As mentioned above, to find the optimal value α^* defined in (14) (or equivalently, θ^* defined in (18)), our plan is to use a stochastic approximation algorithm of the form (15), (16). A key result in stochastic approximation theory (averaging theory), see for example [19], states that under suitable regularity and stability conditions, the behavior of the stochastic approximation algorithm is captured by a *deterministic* ordinary differential equation (ODE) (or, more generally, inclusion) as the step size ϵ goes to zero. Thus to design the stochastic approximation algorithms and give insight into their performance, we will first focus on designing ODEs whose solutions converge to the solution of the constrained MDP (12), (13).In other words, we will construct suitable ODEs whose stable points will be Kuhn-Tucker points of the optimization problem (12), (13).

Once such ordinary differential equations have been designed, the corresponding stochastic approximation algorithms follow naturally by replacing the gradient with the gradient estimate (which is computed from the sample path of the Markov chain), i.e, by replacing $\nabla_\alpha C$, $\nabla_\alpha B$, B in the deterministic algorithms presented below with the estimators $\widehat{\nabla_\alpha C}$, $\widehat{\nabla_\alpha B}$, $\widehat{B}$. These estimates are computed using the parameter free gradient estimation algorithms given in Sec.4.2. The proofs of convergence of the resulting stochastic approximation algorithms are given in Sec.5.

In this section, we present a primal dual and an augmented Lagrangian algorithm. Our technical report [1] also presents a primal algorithm based on gradient projection which requires higher computational complexity.

3.1. *First-Order Primal Dual Algorithm for Constrained MDP*

A widely used deterministic optimization method (with extension to stochastic approximation in [18, pg.180]) for handling constraints is based on the Lagrange multipliers and uses a first-order primal dual algorithm [6, pg 446]. First, convert the inequality constraints (13) in Problem S1 to equality constraints by introducing the variables $z = (z_1, \ldots, z_L) \in \mathbb{R}^L$, so that $B_l(\alpha) + z_l^2 = 0$, $l = 1, \ldots, L$. Define the Lagrangian for the constrained MDP as

$$\mathcal{L}(\alpha, z, \lambda) \stackrel{\triangle}{=} C(\alpha) + \sum_{l=1}^{L} \lambda_l (B_l(\alpha) + z_l^2). \tag{20}$$

Here, $\lambda_l \in \mathbb{R}$, $l = 1, \ldots, L$ are Lagrange multipliers for the constraint $B_l(\alpha) + z_l^2 = 0$. In order to converge, a primal dual algorithm requires the Lagrangian to be locally convex at the the optimum, that is, Hessian to be positive definite at the optimum (which is much more restrictive than the second order sufficiency condition of Assumption 1 in Sec. 2.2). Numerical examples show that this positive definite condition on the Hessian, which Luenberger [20, pp.397] terms "local convexity", seldom holds in the MDP case. We can "convexify" Problem S1 by adding a penalty term to the objective function (12). The resulting problem is:

$$\min_{\alpha \in \boldsymbol{\alpha}^\mu, z \in \mathbb{R}^L} C(\alpha) + \frac{\rho}{2} \sum_{l=1}^{L} \left(B_l(\alpha) + z_l^2 \right)^2,$$

subject to (13). Here ρ denotes a large positive constant. The optimum of the above problem [20, pg.429] is identical to that of (12), (13). Define the augmented Lagrangian,

$$\mathcal{L}_\rho(\alpha, z, \lambda) \stackrel{\triangle}{=} C(\alpha) + \sum_{l=1}^{L} \lambda_l (B_l(\alpha) + z_l^2) + \frac{\rho}{2} \sum_{l=1}^{L} \left(B_l(\alpha) + z_l^2 \right)^2. \tag{21}$$

Note that the original Lagrangian may not be convex near the solution (and hence the primal dual algorithm does not work). for sufficiently large ρ, the last term in $\mathcal{L}_\rho$ "convexifies" the However, for sufficiently large ρ, [6] shows that the augmented Lagrangian is locally convex. After some further calculations detailed in [6, pg.396 and 397], the primal dual algorithm operating on $\mathcal{L}_\rho(\alpha(n), z(n), \lambda(n))$ reads:

$$\alpha^\epsilon(n+1) = \alpha^\epsilon(n) - \epsilon \left(\nabla_\alpha C(\alpha^\epsilon(n)) + \nabla_\alpha B(\alpha^\epsilon(n)) \max \left[0, \lambda^\epsilon(n) + \rho B(\alpha^\epsilon(n)) \right] \right) \tag{22}$$

$$\lambda^\epsilon(n+1) = \max \left[\left(1 - \frac{\epsilon}{\rho} \right) \lambda^\epsilon(n), \lambda^\epsilon(n) + \epsilon B(\alpha^\epsilon(n)) \right] \tag{23}$$

where $\epsilon > 0$ denotes the step size and the notation $z = \max[x, y]$ for any two equal dimensional vectors x, y denotes the vector z with components $z_i = \max[x_i, y_i]$.

Lemma 1. *Under Assumption 1, for sufficiently large $\rho > 0$, there exists $\bar{\epsilon} > 0$, such that for all $\epsilon \in (0, \bar{\epsilon}]$, the sequence $\{\alpha^\epsilon(n), \lambda^\epsilon(n)\}$ generated by the primal dual algorithm (22) is attracted to a local KT pair (α^*, λ^*) of Problem S1.*

Proof. Since $\mathcal{L}_\rho$ is locally convex for sufficiently large $\rho > 0$ [20], the proof straightforwardly follows from Proposition 4.4.2 in [6]. □

Let $T \in \mathbb{R}^+$ denote a fixed constant and $t \in [0, T]$ denote continuous time. Define the piecewise constant interpolated continuous-time process

$$\alpha^\epsilon(t) = \alpha^\epsilon(n) \quad t \in [n\epsilon, (n+1)\epsilon) \tag{24}$$

$$\lambda^\epsilon(t) = \lambda^\epsilon(n) \quad t \in [n\epsilon, (n+1)\epsilon). \tag{25}$$

Lemma 1 implies that $\{\lambda^\epsilon(n)\}$ lies in a compact set Λ for all n. Recall from (8), that $\alpha^\epsilon(n) \in \boldsymbol{\alpha}^\mu$ where $\boldsymbol{\alpha}^\mu$ is compact. Then the following result follows directly from the above lemma and [18] where the convergence of the stochastic version is proved.

Theorem 1. *Let $\{\alpha(t), \lambda(t)\}$ satisfy the ODEs*

$$\frac{d}{dt}\alpha(t) = -\nabla_\alpha C(\alpha(t)) - \nabla_\alpha B(\alpha(t)) \max\Big[0, \lambda(t) + \rho B(\alpha(t))\Big], \quad \alpha(0) \in \boldsymbol{\alpha}^\mu$$

$$\frac{d}{dt}\lambda_l(t) = \begin{cases} B_l(\alpha(t)) & \text{if } \lambda_l(t) + \rho B_l(\alpha(t)) \geq 0 \\ \lambda_l(t)/\rho & \text{if } \lambda_l(t) + \rho B_l(\alpha(t)) < 0, \end{cases} \quad \lambda_l(0) \in \mathbb{R}, \quad l = 1, \ldots, L. \tag{26}$$

Then, under Assumption 1, for sufficiently large ρ (see Lemma 1), the interpolated process $\{\alpha^\epsilon(t), \lambda^\epsilon(t)\}$ defined in (24), (25) converges uniformly as $\epsilon \to 0$ to the process $\{\alpha(t), \lambda(t)\}$, that is,

$$\lim_{\epsilon \downarrow 0} \sup_{0 < t \leq T} |\alpha^\epsilon(t) - \alpha(t)| = 0, \lim_{\epsilon \downarrow 0} \sup_{0 < t \leq T} |\lambda^\epsilon(t) - \lambda(t)| = 0, \tag{27}$$

The attraction point of (26) is the local KT pair (α^*, λ^*) of Problem S1.

3.2. *Augmented Lagrangian (Multiplier) Algorithms for Constrained MDP*

We outline two augmented Lagrangian (multiplier) deterministic algorithms for optimizing the constrained MDP assuming full knowledge of the objective $C(\alpha)$, constraints $B(\alpha)$ and their derivatives.

1. Inexact Primal Minimization Multiplier Algorithm: The augmented Lagrangian approach (also known as a multiplier method) consists of the following coupled ODE and difference equation:

$$\frac{d\alpha^{(n+1)}(t)}{dt} = -\nabla_\alpha C(\alpha^{(n+1)}(t)) - \nabla_\alpha B(\alpha^{(n+1)}(t)) \max\left[0, \lambda(n) + \rho B(\alpha^{(n+1)}(t)\right] \tag{28}$$

$$\lambda_l(n+1) = \max\left[0,\ \lambda_l(n) + \rho B_l(\alpha^{(n+1)}(\infty))\right], \quad l = 1, \ldots, L, \tag{29}$$

where $\alpha^{(n+1)}(\infty)$ denotes the stable point of the ODE (28). Iteration (29) is a first order update for the multiplier, while (28) represents an ODE which is attracted to the minimum of the augmented Lagrangian $\mathcal{L}_\rho$. The $\max$ in (29) arises in dealing with the inequality constraints, see [6, pp.396]. [6, Proposition 4.2.3] shows that if $(\alpha^{(0)}(0), \lambda_0(0))$ lies in the domain of attraction of a local KT pair (α^*, λ^*), then (28), (29) converges to this KT pair. A practical alternative to the above exact primal minimization is first order *inexact minimization* of the primal. The iterative version of the algorithm reads [6, pg.406]: At time $n+1$ set $\alpha^{(0)}(n+1) = \alpha(n)$. Then run $j = 0, \ldots, J-1$ iterations of the following gradient minimization of the primal

$$\alpha^{(j+1)}(n+1) = \alpha^{(j)}(n+1) - \epsilon\Big(\nabla_\alpha C(\alpha^{(j)}(n)) + \nabla_\alpha B(\alpha^{(j)}(n)) \max\left[0, \lambda(n) + \rho B(\alpha^{(j)}(n))\right]\Big), \tag{30}$$

$\alpha(n+1) = \alpha^{(J)}(n+1)$ followed by a first order multiplier step

$$\lambda_l(n+1) = \max\left[0,\ \lambda_l(n) + \rho B_l(\alpha(n+1))\right], \quad l = 1, \ldots, L. \tag{31}$$

Iteration (30) represents a first order fixed step size inexact minimization of the augmented Lagrangian $\mathcal{L}_\rho$ (*inexact* because (30) is terminated after a finite number of steps J). It is shown, see [6] and references therein, that as long as the inexact minimization of the primal is done such that the error tolerances are decreasing with n but summable, then the algorithm converges to a Kuhn Tucker point.

2. Fixed Multiplier: A trivial case of the multiplier algorithm is to fix $\lambda(n) = \bar{\lambda}$ for all time n and only update α according to (30) with $I = 1$ iteration at each time instant. This is clearly equivalent to the primal update (22) with fixed $\lambda(n) = \bar{\lambda}$. From Theorem 1, the interpolated trajectory of this algorithm converges uniformly as $\epsilon \to 0$ to the trajectory of the ODE

$$\frac{d}{dt}\alpha(t) = -\nabla_\alpha C(\alpha(t)) - \nabla_\alpha B(\alpha(t)) \max\left[0, \bar{\lambda} + \rho B(\alpha(t))\right], \quad \alpha(0) \in \boldsymbol{\alpha}^\mu. \tag{32}$$

The following result in [6] shows that the attraction point of this ODE is close to α^* for sufficiently large ρ, resulting in a near optimal solution. First convert the L inequality constraints to equality constraints as outlined in Sec.3.1. Let α^*, λ^* denote the corresponding KT pair.

Result 1. [6, Proposition 4.2.3]. Let $\bar{\rho} > 0$ be scalar such that $\nabla_\alpha{}^2 \mathcal{L}_\rho(\alpha^*, \lambda^*) > 0$. Then there exist positive scalars δ and K such that for $(\bar{\lambda}, \rho) \in D \in \mathbb{R}^{L+1}$ defined by

$$D = \{(\lambda, \rho) : \|\lambda - \lambda^*\| < \delta\rho,\ \rho \geq \bar{\rho}\}$$

the attraction point $\alpha^{\lambda,\rho}$ of the ODE (32) is unique. Moreover, $\|\alpha^{\lambda,\rho} - \alpha^*\| \leq K(\|\bar{\lambda} - \lambda^*\|)/\rho$.

4. Gradient Estimation Algorithms for Constrained MDP

In the previous section, we presented deterministic algorithms for optimizing the constrained MDP Problem S1 that assume complete knowledge of the gradient of the expected cost and constraints. This section deals with how to estimate the gradient of the cost and constraint by simulation. As mentioned earlier, this gradient estimation step is a key step in the stochastic approximation algorithms to optimize the constrained MDP.

Throughout this section we focus on the process with a fixed value of the control parameter α. We use $\mathbb{E}_\alpha$ to denote expectation w.r.t the underlying probability measure of $\{Z_n, n = 1, 2, \ldots\}$. We will also use E_i to denote expectation w.r.t. to the distribution of the random action, given that the state is $X_n = i$. Finally, Z_n will be refereed to as the "*nominal process*."

4.1. *Infinite Horizon Measure Valued Derivative Estimation*

Although we are interested in estimating the gradient given short data lengths (so as to deal with time varying transition probabilities), to fix ideas, we first focus on the infinite horizon case, that is, we build gradient estimators of

$$\nabla_\alpha \mathbb{E}_{\pi(\alpha)}[F(Z)] = \nabla_\alpha \left(\sum_{i \in S} \sum_{u \in \mathcal{U}_i} F(i,u)\pi_{i,u}(\alpha) \right) \tag{33}$$

Here F indicates a function of the state-action pair $Z = (i, u)$, such as the cost c or the constraint values b_l, and $\pi_{i,u})\alpha)$ is the stationary measure of the chain $\{Z_n\}$ under fixed parameter value α. The resulting estimators estimators $\widehat{\nabla_\alpha C}$ and $\widehat{\nabla_\alpha B}$ are parameter free (learning) gradient estimators.

In the case of spherical coordinates, it follows from the relationship (6) that the action u_{k+1} (given $i_{k+1} = i$) has a distribution

$$u_{k+1} = \begin{cases} 0 & \text{w.p. } \cos^2(\alpha_{i1}) \\ Y_1 & \text{w.p. } \sin^2(\alpha_{i1}) \end{cases} \tag{34}$$

$$Y_1 = \begin{cases} 1 & \text{w.p. } \cos^2(\alpha_{i2}) \\ Y_2 & \text{w.p. } \sin^2(\alpha_{i2}) \end{cases}$$

$$\vdots$$

$$Y_{d(i)-1} = \begin{cases} d(i)-1 & \text{w.p. } \cos^2(\alpha_{i,d(i)}) \\ d(i) & \text{w.p. } \sin^2(\alpha_{i,d(i)}). \end{cases}$$

Let $Y_{d(i)} = d(i)$. The value of α_{ip}, $i \in S$, $p \in \{1, 2, \dots d(i)\}$ does not affect the distribution of u_{k+1} if $i_{k+1} \neq i$, therefore the gradient of the one-step transition expectation is non null only when $i_{k+1} = i$, in which case we have

$$\frac{\partial}{\partial \alpha_{ip}} \mathbb{E}[f(i, u_{k+1})] = \frac{\partial}{\partial \alpha_{ip}} \mathbb{E}\Big[f(i, p-1)\cos^2(\alpha_{ip}) + f(i, Y_p)\sin^2(\alpha_{ip}) \Big] \prod_{k=1}^{p-1} \sin^2(\alpha_{ik})$$

$$= -2\sin(\alpha_{ip})\cos(\alpha_{ip}) \prod_{k=1}^{p-1} \sin^2(\alpha_{ik}) \mathbb{E}[f(i, p-1) - f(i, Y_p)]$$

because the terms $f(i,k), k < p-1$ have weights which are independent of α_{ip}. The random variable Y_p is called the "phantom action" and it has a distribution concentrated on $\{p, \dots, d(i)\}$ corresponding to

$$\mathbb{P}(Y_p = a) = \frac{\theta_{ia}(\alpha)}{\prod_{m=1}^{p-1} \sin^2(\alpha_{im})}, \quad a \geq p,\ p \in \{1, 2, \dots d(i)\}. \tag{35}$$

Note that by construction, for $p = d(i)$ the random variable $Y_{d(i)} = d(i)$ is degenerate.

The random variable Y_p is called the "phantom action" and it has a distribution concentrated on $\{p, \dots, d(i)\}$ corresponding to

$$\mathbb{P}(Y_p = a) = \frac{\theta_{ia}(\alpha)}{\prod_{m=1}^{p-1} \sin^2(\alpha_{im})}, \quad a \geq p,\ p \in \{1, 2, \dots d(i)\}. \tag{36}$$

Note that by construction, for $p = d(i)$ the random variable $Y_{d(i)} = d(i)$ is degenerate.

The above analysis for the one-step transition expectation is an example of what is called a "weak derivative" [23]. To obtain the gradient of an infinite horizon expectation (33), a measure-valued approach for weak derivatives is used in [1]. In the proof of Theorem 2, we will use the approach of the perturbation realization factors. The equivalence between the two approaches has been shown in [12].

First introduce the following notation. Let $\{\tilde{u}_k\}$ denote a sequence of iid random variables (independent of $\mathfrak{F}_n$) as follows:

$$\tilde{u}_k = \begin{cases} Y_p, & p < d(i) \\ d(i)\mathbf{1}_{\{u_k = d(i)-1\}} + (d(i)-1)\mathbf{1}_{\{u_k = d(i)\}} & p = d(i) \end{cases} \tag{37}$$

we now provide the basis for the interpretation of $\tilde{u}_k$. For $p-1 < d(i)$ $\mathbb{P}(u_{k+1} = p-1 \mid X_{k+1} = i) = \cos^2(\alpha_{ip}) \prod_{k=1}^{p-1} \sin^2(\alpha_{ik})$, so that we can re write the derivative expression as:

$$\frac{\partial}{\partial \alpha_{ip}} \mathbb{E}[f(i, u_{k+1})] = -2\tan(\alpha_{ip}) \mathbb{P}(u_{k+1} = p-1 \mid X_{k+1} = i) \mathbb{E}[f(i, u_{k+1}) - f(i, \tilde{u}_{k+1})] \mid u_{k+1} = p-1],$$

which is the basis for using an 'on-line" estimator, as we will shortly explain in detail: when the state is $(X_{k+1}, u_{k+1} = (i, p-1)$, we calculate a difference between the observed trajectory and that of the phantom process, where the only difference is that the phantom action is now $\tilde{u}_k$. Because the case $p = d(i)$ has a degenerate distribution for Y_p, the estimation can be made both when $u_{k+1} = d(i)$ (in which case the phantom action is $d(i)-1$), or when it is $d(i)-1$ (in which case the phantom action is $d(i)$). The complete details of this reasoning appears in [1].

Define the hitting time

$$\nu(k) = \min\{n > 0 : Z_{k+n} = (i, \tilde{u}_k)\}. \tag{38}$$

Theorem 2. *Consider the MDP $\{Z_n\}$ governed by (5) and (9) with $\alpha \in \boldsymbol{\alpha}^\mu$. Fix (i,p), $a = p-1$. Let $\hat{\Gamma}$ denote any estimator that converges as $N \to \infty$ to $C(\alpha)$ a.s. (e.g., $\hat{\Gamma} = \frac{1}{N}\sum_{n=1}^{N} c(Z_n)$) and $K_{ip}(\alpha, k)$ defined as:*

$$K_{ip}(\alpha, k) = \begin{cases} -\tan(\alpha_{ip})\, \delta_{i,p}(k) & \text{if } p = 1, \ldots, d(i)-1 \\ -\cos(\alpha_{i,d(i)}) \sin(\alpha_{i,d(i)}) [\delta_{i,d(i)}(k) - \delta_{i,d(i)+1}(k)] & \text{if } p = d(i) \end{cases} \tag{39}$$

Here $\delta_{ip}(k) = \mathbf{1}_{\{Z_k = (i,p-1)\}}$.

Then for fixed $\alpha \in \boldsymbol{\alpha}^o$, a parameter free consistent estimator for the gradient

in (33) is given by:

$$\hat{G}_N(i,p) = \frac{2}{N}\sum_{k=1}^{N} K_{ip}(\alpha,k)\left([c(i,u_k) - c(i,\tilde{u}_k)] + \sum_{n=k+1}^{k+\nu(k)} c(Z_n) - \nu(k)\,\hat{\Gamma}\right), \quad p = 1,\ldots,d(i). \tag{40}$$

The proof of the above theorem is in the appendix. The algorithm computes the differences between the *nominal* (observed) process trajectory and the "parallel" ones that start at time k with action $\tilde{u}_k$. Such parallel processes are called "phantom" processes, and we call our algorithm the *WD phantom estimator*.

4.2. *Fast WD for Tracking Time-Varying MDPs*

In Theorem 2, consistency of the WD phantom estimator for large sample size N was established. However, for adaptive control of constrained MDPs with time varying transition probabilities, it is necessary to implement the estimation over small batch sizes of the observed system trajectory so that the iterates of the stochastic gradient algorithm are performed more frequently to track the optimal time varying α^*. The aim of this section is to present an implementation of the WD phantom estimators over short batch sizes N and to show that the resulting gradient estimate is still consistent. We call these "fast WD phantoms".

The implementation of the fast WD phantoms over short batch sizes proceeds as follows. Suppose that the gradient $\nabla_\alpha C(\alpha)$ is to be estimated using the observed MDP trajectory over the nth batch $I_n \equiv \{nN+1,\ldots,(n+1)N\}$. As in Theorem 2, let $\widehat{\Gamma}_n$ denote an estimator of $C(\alpha)$ using the observed trajectory of the MDP in I_n. The fast WD phantom estimators for the components (i,p), $i \in S$, $p \in \{1,2,\ldots d(i)\}$, of the gradient are (compare with (40)):

$$\widehat{\hat{G}}_n(i,p) = \frac{2}{N}\Bigg[\sum_{k=nN+1}^{(n+1)N} K_{ip}(\alpha,k)\Bigg[c(i,u_k) - c(i,\tilde{u}_k) + \sum_{j=k+1}^{\min\{(k+\nu(k)),(n+1)N\}} c(Z_j) - \nu(k)\widehat{\Gamma}_n\mathcal{D}_n(k)\Bigg]$$
$$+ \sum_{k\in\mathcal{L}(nN)} K_{ip}(\alpha,k)\Bigg(\sum_{j=nN+1}^{\min\{(k+\nu(k)),(n+1)N\}} c(Z_j) - \nu(k)\widehat{\Gamma}_n\mathcal{D}_n(k)\Bigg)\Bigg] \tag{41}$$

where $\mathcal{D}_n(k) = \mathbf{1}_{\{k+\nu(k)\in I_n\}}$ (phantom k dies in I_n), and $\mathcal{L}(j) = \{k \le j\colon \nu(k)+k > j\}$ is the list of living phantoms at stage j. A similar estimator holds for the gradient of the constraints.

The interpretation of (41) is as follows. At each step k, the state $Z_k = (i,a)$ is observed and a new phantom system (labelled by k) is started, generating the phantom decision $\tilde{u}_k$ as described above. The k-th phantom system "dies" at the

hitting time $j = k+\nu(k)$, otherwise it is contained in the set of "living" phantoms $\mathcal{L}(j)$. In a computer program, this corresponds to a list. This phantom will be used to estimate the partial derivative of all the functions (cost and constraints) with respect to $\alpha_{i,a+1}$ if $a < d(i) - 1$. If $a = d(i) - 1$ or $a = d(i)$, the corresponding phantom system contributes (with opposite signs) to the estimation of the gradient w.r.t. $\alpha_{i,d(i)}$. The difference in costs inside the brackets in (41) contains the initial contribution of a phantom system. Afterwards, while a phantom system k is alive, it contributes to (41) the term $c(Z_j)$ at each step, and when it dies ($\mathcal{D}_n(k) = 1$) it contributes the term $\nu(k)\widehat{\Gamma}_n$ (if death occurs within the interval I_n). The above equation takes only observations of the trajectory within the *current* interval, thus a final term appears considering the contributions of all the living phantoms at the start of the interval, because it is possible that phantom systems may survive several estimation intervals. For complete details on the implementation program code and other variations please see [17].

Theorem 3. *Assume that for any $\alpha \in \boldsymbol{\alpha}^o$, $\mathbb{E}_{\pi(\alpha)}\widehat{\Gamma}_n = C(\alpha)$. Then the bias of the fast WD phantom gradient estimator $\widehat{\widehat{G}}_n(i,p)$ of (41) per batch of size N is given by (with $K_{ip}(\alpha,k)$ defined in (39))*

$$\mathbb{E}_{\pi(\alpha)}[\widehat{\widehat{G}}_n(i,p)] - \nabla_\alpha C(\alpha) = \frac{2}{N} \sum_{k:\ k+\nu(k)\in I_n} K_{ip}(\alpha,k)\, \mathrm{Cov}_{\pi(\alpha)}(\nu(k), \widehat{\Gamma}_n) + o(\frac{1}{N}). \tag{42}$$

Proof. Let $\widehat{\widehat{G}}^o_n(i,p)$ denote the gradient estimate (41) when $\hat{\Gamma}_N$ is replaced by $C(\alpha)$. We first show that the gradient estimate $\widehat{\widehat{G}}^o_n(i,p)$ is unbiased under the invariant measure $\pi(\alpha)$, that is, $\mathbb{E}_{\pi(\alpha)}[\widehat{\widehat{G}}^o_n(i,p)] = \nabla_\alpha C(\alpha)$.

Note that under $\pi(\alpha)$, consecutive estimators have the same distribution (although they are not independent), because the invariant distribution of the number of living phantoms at the start of the interval is independent of n. From the ergodicity of the underlying MDP, $\mathbb{E}_{\pi(\alpha)}[\widehat{\widehat{G}}^o_n(i,p)] = \lim_{n\to\infty}(1/n)\sum_{m=0}^{n-1}\widehat{\widehat{G}}^o_m(i,p)$. Because the estimation by batches considers breaking up the partial sums of the estimation, then the difference:

$$\frac{1}{n}\sum_{m=0}^{(n-1)}\widehat{\widehat{G}}^o_m(i,p) - \frac{2}{nN}\sum_{k=1}^{nN} K_{ip}(\alpha,k)\left[c(i,a) - c(i,\tilde{u}_k) + \sum_{j=(k+1)}^{k+\nu(k)} c(Z_j) - \nu(k)C(\alpha)\right]$$

tends to zero in absolute value, a.s. Using Theorem 2, the $\frac{2}{nN}\sum_k$ term in the above equation converges a.s. to $\nabla_\alpha C(\alpha)$ as $n \to \infty$, for any fixed value of the batch size N. This establishes the claim that the gradient estimate is unbiased.

Next, using the above expression consider $\widehat{\widehat{G}}_n^o - \widehat{\widehat{G}}_n$. Then using the fact that the consecutive estimators are additive, we obtain:

$$\frac{1}{n}\sum_{m=0}^{n-1}(\widehat{\widehat{G}}_n^o - \widehat{\widehat{G}}_n) = \frac{1}{n}\sum_{m=0}^{n-1}\left[\frac{2}{N}\sum_{k=1}^{nN}\nu(k)(\hat{\Gamma}_m - C(\alpha))\mathbf{1}_{\{k+\nu(k)\in I_m\}}\right] + e(nN).$$

The error term $e(nN)$ comes from the phantom systems $k\colon k+\nu(k) \le nN$, and their contribution tends to zero a.s. as $n \to \infty$. Under the invariant measure, we have:

$$\mathbb{E}_{\pi(\alpha)}\left(\nu(k)(\hat{\Gamma}_m - C(\alpha))\mathbf{1}_{\{k+\nu(k)\in I_m\}}\right) = \mathrm{Cov}_{\pi(\alpha)}(\nu(k)\mathbf{1}_{\{k+\nu(k)\in I_m\}}, \hat{\Gamma}_m) + C(\alpha)\mathbb{E}_{\pi(\alpha)}(\nu(k)\mathbf{1}_{\{k+\nu(k)\in I_m\}}).$$

□

4.3. *Computational Cost and Memory Requirement*

Consider the fast WD phantom algorithm. Let $\alpha \in \boldsymbol{\alpha}^\mu$. We will bound stochastically the number of living phantoms in terms of Binomial random variables.

Consider the process $\{Z_n\}$ in stationary operation, and call $\mathcal{L}_{nN}(i,a)$ the number of living phantoms in (41) that are waiting to hit state (i,a). All these phantoms were created at some earlier time instant when the chain hit the state $Z_k = (i,p)$ and the phantom decision was $\tilde{u}_k = a$. Clearly, the maximum number of phantoms in $\mathcal{L}_{nN}(i,a)$ satisfies:

$$\|\mathcal{L}_{nN}(i,a)\| = \sum_{a>p}\sum_{k=t(i,a)}^{nN}\mathbf{1}_{\{X_k=i;u_k=p,\tilde{u}_k=a\}},$$

where $t(i,a) = \max(j \le nN\colon X_j = 1, u_j = a)$, because if the state (i,a) is visited at time j, at that time all living phantoms that were in $\mathcal{L}_{j-1}(i,a)$ die and $\mathcal{L}_j(i,a)$ is empty. Call $\tau(i,a)$ the return time to state (i,a) and let $n(i)$ be the number of visits to state i ($X_k = i$) within two consecutive visits to state (i,a). Then the number of phantom systems in $\mathcal{L}_{nN}(a)$ is bounded by a Binomial$(\mathbf{p}(a), n(i))$, where

$$\mathbf{p}(a) = \frac{\theta_{ia}}{\sum_{m=1}^{p-1}\sin^2(\alpha_{im})},$$

according to the creation of the phantom systems. Clearly considering the maximum value of all such probabilities, we can bound the number of phantoms on each list for every value of $\alpha \in \alpha^\mu$.

In (41), the estimator $\widehat{\widehat{G}}_n^o(i,p)$ is composed of bounded quantities (the state space is finite) plus a contribution of N terms of the order of $\nu(k)$ each, plus a

contribution which is proportional to the random variable:

$$\sum_{a>p}\sum_{k\in\mathcal{L}_{nN}(i,a)}\nu(k)\leq\sum_{a>p}\sum_{i=1}^{n(i)}\nu(k)\leq\sum_{a>p}\sum_{k=1}^{\tau(i,a)}\nu(k)$$

Note that, as in the proof of Theorem 2, $\nu(k)<\max(\tau(i,a):a>p)=\tau$ a.s. Because α^{μ} is a compact set with ergodic states, the return times are all finite a.s.. and $\mathbb{E}_{\alpha}[\tau(i,a)]=1/\pi_{ia}$. Therefore, boundedness of the m-th moment of $\widehat{\widehat{G}}_n^o(i,p)$ now follows from boundedness of the $2m$-th moment of τ.

5. Stochastic Gradient Algorithms for Constrained MDP

In this section we present the stochastic gradient algorithms that use the parameter free gradient estimators (fast WD phantoms) of Sec.4.2 to optimize the constrained MDP given in Problem S1 ((12), (13)). Also weak convergence proofs of these algorithms are presented. The stochastic algorithms presented are stochastic versions of the two deterministic algorithms of Sec.3.1 and Sec.3.2. Using the simulation-based fast WD phantom gradient estimator (41) with local sample averages for the estimation of the constraint functions may lead to a noticeable bias, for small batch size N thus making the algorithm suboptimal.

For notational convenience we consider equality constraints here (as mentioned in Sec.3.1 the inequality constraints can be handled with minor modifications) so that the constrained MDP problem (12), (13) reads

$$\min_{\alpha\in\boldsymbol{\alpha}^{\mu}}C(\alpha),\quad\text{s.t. }B_l(\alpha)=0,\quad l=1,\ldots,L$$

A control "agent" is associated with each of the possible visited states $i\in S$. The control parameter for this agent is the vector $(\alpha_{ip},p\in\{1,\ldots,d(i)\})$, plus an agent for the (artificial control) variable representing the Lagrange multiplier λ. The scheme works by observing the process over a batch size N during which the value of the control parameter does not change. Over this batch, the constraint is estimated as $\hat{B}(n)$, see Sec.5.3, and the gradients are estimated as $\widehat{\nabla_{\alpha}C}(n)$, $\widehat{\nabla_{\alpha}B}(n)$ using (41). We assume that $\mathbb{E}_{\pi(\alpha)}[\hat{B}(n)]=B(\alpha)$. Let

$$h_0(\alpha)=\mathbb{E}_{\pi(\alpha)}[\widehat{\nabla_{\alpha}C}(n)],\quad h_l(\alpha)=\mathbb{E}_{\pi(\alpha)}[\widehat{\nabla_{\alpha}B_l}(n)],\quad l=1,\ldots,L$$

be the invariant averages of the batch estimation (refer to Theorem 3).

5.1. *Stochastic First-Order Primal Dual Algorithm*

Consider the stochastic approximation where the control parameter (α, λ) is updated as (c.f. (22), (23))

$$\alpha^\epsilon(n+1) = \alpha^\epsilon(n) - \epsilon \left(\widehat{\nabla_\alpha C}(n) + \sum_{l=1}^{L} \left(\lambda_l^\epsilon(n) + \rho \hat{B}_l(n) \right) \widehat{\nabla_\alpha B}_l(n) + Z^\mu(n) \right), \quad \alpha^\epsilon(0) \in \boldsymbol{\alpha}^\mu \tag{43}$$

$$\lambda^\epsilon(n+1) = \lambda^\epsilon(n) + \epsilon \hat{B}(n) \tag{44}$$

where the gradient estimators are given by the fast WD phantoms in (41) and $Z^\mu(n)$ is a projection that ensures that $\{\alpha^\epsilon(n)\} \in \boldsymbol{\alpha}^\mu$. The above truncation is a mathematical artifice to prove convergence. In practical implementation, that is, when $\epsilon > 0$, truncation is not important since one can choose $\mu << \epsilon$, e.g., close to the numerical resolution of the computer, see remark below.

Proposition 1.1. *For $\alpha^\epsilon(n), \lambda^\epsilon(n)$ given by (43), (44), define the interpolated process $\{\alpha^\epsilon(t), \lambda^\epsilon(t)\}$ as in (24), (25). Then as $\epsilon \to 0$, $\{a^\epsilon(t), \lambda^\epsilon(t)\}$ converges in distribution to the solution of the ODE (c.f. (26))*

$$\frac{d}{dt}\alpha(t) = -\left[h_0[\alpha(t)] + \sum_{l=1}^{L} (\lambda_l(t) + \rho\, B_l[\alpha(t)])\, h_l[\alpha(t)] + \kappa[\alpha(t)] \right]_\mu, \quad \alpha(0) \in \boldsymbol{\alpha}^\mu \tag{45}$$

$$\frac{d}{dt}\lambda(t) = B[\alpha(t)],$$

where the notation $[\cdot]_\mu$ refers to the truncated ODE onto $\boldsymbol{\alpha}^\mu$ and the added drift is defined as

$$\kappa(\alpha) = \rho \lim_{n\to\infty} \sum_{l=1}^{L} \mathrm{Cov}_{\pi(\alpha)}[\hat{B}_l(n), \widehat{\nabla_\alpha B_l}(n)]. \tag{46}$$

Remark 1.3. A weak convergence proof of the stochastic gradient algorithm (43), (44) requires uniform integrability of the gradient estimates. Without the above truncation, if $\alpha_{ip} = \pi/2$ or $\alpha_{ip} = 0$, (equivalently one or more action probabilities $\theta_{iu} = 0$), then the hitting time $\nu(k)$ in (38) of the phantom system k with initial state (i, u) is not uniformly bounded and the gradient estimator (41) is not well defined. So in the weak convergence proof below we place a μ size ball around the boundary of $\boldsymbol{\alpha}$ (recall $\boldsymbol{\alpha}^\mu$ above is defined as $\boldsymbol{\alpha}$ minus this ball) and the estimates $\alpha^\epsilon(n)$ are truncated to $\boldsymbol{\alpha}^\mu$. However, this truncation is very different to standard truncations in the stochastic approximation literature:

1. Recall from Sec.2.1 that the boundary of the set $\boldsymbol{\alpha}$ is a fictitious boundary and the action probabilities $\theta(\alpha)$ are symmetric about this fictitious boundary (since they are functions of $\sin^2(\alpha)$ and $\cos^2(\alpha)$). For example, suppose that $\alpha^{\epsilon}_{ip}(n) < \pi/2 - \mu$ and that (43) generates the estimate $\alpha^{\epsilon}_{ip}(n+1) = \pi/2 + \mu + K$ for some small constant $K > 0$. Then truncation is not required since by symmetry $\alpha^{\epsilon}_{ip}(n+1) = \pi/2 - (\mu + K) \in \boldsymbol{\alpha}^{\mu}$. Thus in practical implementation, that is, when $\epsilon > 0$, truncation is not important since we can choose $\mu << \epsilon$. The probability that an update lies precisely in a ball of radius μ is negligibly small – since if the estimate overshoots or undershoots this ball, it is automatically in $\boldsymbol{\alpha}^{\mu}$.
2. Suppose that the untruncated version of the ODE (45) has a stable point on the boundary of $\boldsymbol{\alpha}$, e.g., a pure policy. Then clearly, the truncated ODE will have a stable point within $O(\mu)$ of the stable point of the untruncated ODE. Hence the truncation is very different to standard ODE truncations such as (26).

Proof. We first show that the term in parenthesis on the RHS of (43) is uniformly integrable. Note that all the terms in the gradient estimate (41) apart from $\mathcal{L}(\cdot)$ and $\nu(\cdot)$ are uniformly bounded for $\alpha \in \boldsymbol{\alpha}^{\mu}$ since the MDP is finite state and the batch size N is fixed and finite. Hence a sufficient condition for uniform integrability is to show that $\sum_{k \in \mathcal{L}(nN)} \nu(k)$ in (41) has finite variance.

Fix (i,p) for the estimator in (41). We focus on the phantoms that have an initial decision $\tilde{u}_k = a$ for a fixed a – and we call these a-phantoms. By definition, all a-phantoms die simultaneously at time n_1 when the process $Z_{n_1} = (i,a)$. Let n_0 denote the previous time instant at which $\{Z_n\}$ visited (i,a), that is, $Z_{n_0} = (i,a)$. The longest hitting time $\nu(k)$ of all these phantoms is bounded by the time between successive returns $n_1 - n_0$. Consider now the number of living a-phantoms in $\mathcal{L}(n)$ at any time $n \in [n_0, n_1]$. Each of these a-phantoms must have been created when the process hits the state (i,p) and the phantom decision chosen is a – which happens with probability $\frac{\theta_{ia}}{\sum_{m \geq a} \theta_{im}}$. Therefore an almost sure upper bound for the cardinality of $\mathcal{L}(n)$ is $n_1 - n_0$. Note that $n_1 - n_0$ is the return time of a ergodic MDP on a finite state and therefore has all moments bounded.

Having established uniform integrability of the updates, the result follows by direct application of Theorem 5.2.1 in [18]. The continuity of the invariant expectations follows from the fact that the transition kernel of Z_n is analytic in α. To characterize the drift functions, use:

$$\mathbb{E}_{\pi(\alpha)}\left[\hat{B}_l(n)\widehat{\nabla_\alpha B}_l(n)\right] = \mathbb{E}_{\pi(\alpha)}[\hat{B}_l(n)]\,\mathbb{E}_{\pi(\alpha)}[\widehat{\nabla_\alpha B}_l(n)] + \mathrm{Cov}_{\pi(\alpha)}[\hat{B}_l(n), \widehat{\nabla_\alpha B_l}(n)],$$

which establishes the result. □

5.2. *Stochastic Augmented Lagrangian Multiplier Algorithm*

Consider the following stochastic approximation version of the multiplier algorithm (31):

$$\alpha^\epsilon(n+1) = \alpha^\epsilon(n) - \epsilon \left(\widehat{\nabla_\alpha C}(n) + \sum_{l=1}^{L} \big(\lambda_l^\epsilon(n) + \rho \hat{B}(n)\big) \widehat{\nabla_\alpha B_l}(n) + Z^\mu(n) \right), \quad \alpha^\epsilon(0) \in \boldsymbol{\alpha}^\mu \tag{47}$$

where $Z^\mu(n)$ is a projection that ensures that $\{\alpha^\epsilon(n)\} \in \boldsymbol{\alpha}^\mu$ and $\{\lambda^\epsilon(n), \epsilon > 0, n \in \mathbb{N}\}$ is any tight sequence. A trivial example is when $\lambda^\epsilon(n)$ is a bounded constant (a.s.).

The following result regarding the weak convergence of (47) is proved in the appendix.

Proposition 1.2. *Assume that $\{\lambda^\epsilon(n), \epsilon > 0, n \in \mathbb{N}\}$ is tight. Define the interpolated process $\alpha^\epsilon(t)$ of (47) as in (24). Then as $\epsilon \to 0$, the interpolated process $\alpha^\epsilon(t)$ converges in distribution to the solution of the ODE:*

$$\frac{d\alpha(t)}{dt} = - \left[h_0[\alpha(t)] + \sum_{l=1}^{L} (\bar{\lambda}_l + \rho\, B_l[\alpha(t)])\, h_l[\alpha(t)] + \kappa[\alpha(t)] \right]_\mu, \quad \alpha(0) \in \boldsymbol{\alpha}^\mu, \tag{48}$$

where $\bar{\lambda}_l$ is an accumulation point of the sequence $\{\bar{\lambda}(\epsilon), \epsilon > 0\}$ of (convergent) Cesaro sums:

$$\bar{\lambda}(\epsilon) \equiv \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \lambda^\epsilon(n),$$

and $\kappa(\alpha)$ is defined in (46).

Remark 1.4. If the bias in $h_l(\alpha)$ and $\kappa(\alpha)$ are negligible, then under no truncation, the ODE (48) reduces to (32) – which is the ODE for the deterministic fixed multiplier algorithm. Result 1 of Sec.3.2 implies that the estimates converge weakly to a near optimal point, provided that the pair $(\bar{\lambda}, \rho)$ is well chosen. The bias in $h_l(\alpha)$ and $\kappa(\alpha)$ is of order $O(1/N)$.

Consider the following update of the multiplier λ in (47). Define $\mathcal{I} = \lfloor 1/\epsilon \rfloor$ and consider the recursion

$$\lambda(n+1) = \lambda(n) + \bar{B}(n/\mathcal{I}) \mathbf{1}_{\{\frac{n}{\mathcal{I}} \in \mathbb{N}\}}, \quad \text{where } \bar{B}(n/\mathcal{I}) = \frac{1}{\mathcal{I}} \sum_{j=(n-1)\mathcal{I}+1}^{n\mathcal{I}} \hat{B}(j), \tag{49}$$

together with (47). Thus the multiplier is updated once every $\mathcal{I}$ time points. If the bias in $h_l(\alpha)$ and $\kappa(\alpha)$ is negligible, then as $\epsilon \to 0$, the algorithm (47)–(49) converges weakly to the deterministic system (28), (29) which is the exact multiplier algorithm. As mentioned in Sec.3.2, this in turn converges to a local KT point. In a practical implementation, one would choose $\mathcal{I}$ as a large positive integer. In our numerical examples, see [1], even a choice of $\mathcal{I} = 10$ resulted in convergence to a KT point.

5.3. *Tradeoff between Bias and Tracking Ability*

The three sources of bias in the stochastic gradient algorithm (43), (44) are the bias in the estimates $\widehat{\nabla_\alpha B}$, $\widehat{\nabla_\alpha C}$ and $\hat{B}$. A quick mathematical artifice for eliminating the bias is to use batch sizes $N(\epsilon) \to \infty$ as $\epsilon \to 0$. Then the ODE (45) becomes identical to (26). In the numerical examples of [1] we chose $N = 1000$ – for finite N the bias is $O(1/N)$. Although choosing $N(\epsilon) \to \infty$ is theoretically appealing, it is of no practical use since the stochastic gradient algorithm will not respond quickly to changes in the optimal policy caused by time variations in the parameters of the MDP. In our conference paper [17] we use batch sizes $N = 5, 10$ to update the parameter $\alpha(n)$ frequently, and indeed the stochastic approximation algorithm can be implemented even for $N = 1$. The bias in the gradient estimator of Theorem 3 can be controlled using averaging of the estimation of the cost function, or other smoothing statistical techniques. However, what we really are interested in is the resulting bias of the stable point of the limiting ODE. The results of extensive numerical studies indicate that the bias in $h_l(\alpha)$, $l = 0, 1, \ldots, L$, has negligible effect on the behaviour of the stochastic gradient algorithm even for small batch sizes of $N = 5$. For example, we performed comparisons using the local sample average $\hat{\Gamma}_n = \sum_{k \in I_n} c(Z_k)/N$ and then using the actual theoretical value $\hat{\Gamma}_n = C(\alpha)$ in $\widehat{\widehat{G}}_n$ of (41) as well as for the constraints with no remarkable difference in the stochastic gradient algorithm.

The main source of bias for small batch sizes is that introduced by $\kappa(\alpha)$, in Propositions 1.1 and 1.2. Using the local sample average over the n-th batch

$$\hat{B}_l^\epsilon(n) = \widehat{\widehat{B}}_l(n) \stackrel{\triangle}{=} \frac{1}{N} \sum_{m \in I_n} \beta_l(Z_m) \tag{50}$$

yields a noticeable asymptotic bias $\kappa(\alpha)$. A better alternative is to use the Cesaro sum $\hat{B}_l^\epsilon(n) = \frac{1}{n} \sum_{k=1}^n \widehat{\widehat{B}}_l(k)$. Since $\hat{B}_l^\epsilon(n) \to B_l(\alpha)$ a.s. for all $\alpha \in \boldsymbol{\alpha}^o$ as $n \to \infty$, this Cesaro sum estimator would correct the asymptotic bias of the stochastic gradient algorithm. However, running averages do not respond to changes in the underlying parameters (e.g. transition probabilities) of the MDP

since they are decreasing step size algorithms. Hence they cannot be used for tracking time varying optimal policies. To handle this tracking case, we use in [17] an exponential smoothing

$$\alpha^\epsilon(n+1) = \alpha^\epsilon(n) + \epsilon \left(\widehat{\nabla_\alpha C}(n) - \sum_{l=1}^{L} \left(\lambda_l^\epsilon(n) + \rho \hat{B}_l^\epsilon(n) \right) \widehat{\nabla_\alpha B}_l(n) \right) \quad (51)$$

$$\lambda_l^\epsilon(n+1) = \lambda^\epsilon(n) + \epsilon\, \hat{B}_l^\epsilon(n), \quad \hat{B}_l^\epsilon(n+1) = \hat{B}_l^\epsilon(n) + \delta \left(\widehat{\widehat{B}}_l(n) - \hat{B}_l^\epsilon(n) \right), \quad l = 1, \ldots, L, \quad (52)$$

where $\delta > 0$ and $\widehat{\widehat{B}}_l(n)$ is the local sample average in (50). Using a two time scale stochastic approximation argument it can be shown that if $\epsilon/\delta \to 0$, e.g., if $\delta = \sqrt{\epsilon}$, and $\epsilon \to 0$, then the asymptotic limit points of the corresponding ODE are unbiased. In practical implementation, for non zero δ, the estimates are biased. While the asymptotic bias can be controlled, the exponential smoothing delays the reaction time of the stochastic gradient algorithm since a faster time scale has been introduced, as illustrated in Section 7.2.

6. Case Study: Monotone Policies for Packet Transmission Scheduling over Correlated Wireless Fading Channels

In this section we consider a special case of Problem S1 that arises in transmission scheduling in wireless telecommunication systems. The constrained MDP we consider, models a transmission scheduling problem in a wireless telecommunication network. The action set is $\mathcal{U} = \{0, 1\}$ corresponding to *transmit* and *do not transmit*, respectively. By using a Lagrangian formulation for dynamic programming, we show that the optimal policy is a randomized mixture between two deterministic monotone (threshold) policies; such a policy is a two-step staircase function as plotted in Fig. 1. So the weak derivative based stochastic approximation algorithms presented above can be used to estimate this structured optimal policy. Because of the threshold structure of the optimal policy, the algorithm implementation is very efficient.

Consider the following transmission scheduling problem over a correlated fading wireless channel. At each time slot, a user has to decide whether to transmit a packet unless the packet storage buffer is empty. The objective is to minimize the infinite horizon average transmission cost subject to a constraint on the average delay penalty cost. As in [31; 34], we model the correlated fading wireless channel by a finite state Markov chain (FSMC). That is, we assume the channel state evolves according to a FSMC, and the channel state realization is

known at every time slot. At time $n = 0, 1, \ldots$, the system state is the 3-tuple $X_n = [X_n^b, X_n^c, X_n^y]$, where:
(i) $X_n^b \in \mathcal{B} = \{0, 1, \ldots\}$ is the buffer occupancy state
(ii) X_n^c is the state of the correlated wireless communication channel. Assume $X_n^c \in \mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$, where $\mathcal{C}$ is the finite channel state space and $\mathcal{C}_i$ corresponds to a better channel state than $\mathcal{C}_j$ for all $i > j$; X_n^c evolves as a Markov chain with the transition probabilities $A^c(X_{n+1}^c = \mathcal{C}_j | X_n^c = \mathcal{C}_i) = a_{ij}^c$.
(iii) X_n^y is the number of new packets arriving at the buffer. For simplicity, assume an i.i.d binary packet arrival process, that is at any time n $X_n^y \in \mathcal{Y} = \{0, 1\}$, which denotes either the arrival of no packet or one packet, with the probability mass function $P(X_n^y = 1) = \delta$ and $P(X_n^y = 0) = 1 - \delta$.

The system state space is then the countable set $\mathcal{S} = \mathcal{B} \times \mathcal{C} \times \mathcal{Y}$

Let the action sets be $\mathcal{U} = \{0, 1\}$ for every buffer occupancy state $X^b > 0$ and $\mathcal{U}_0 = \{0\}$ for the buffer occupancy state $X^b = 0$, where 0 and 1 correspond to the action of not transmitting and transmitting respectively.

We now define the costs and constraints of constrained MDP in Problem S1:

- The transmission cost function $c(\cdot, \cdot) : \mathcal{C} \times \mathcal{U} \to \mathbb{R}_+$ is a function of the channel state. Assume that when a transmission is not attempted, no transmission cost is incurred, that is $c(\cdot, 0) = 0$.
- The constraint is specified by a delay penalty cost $\beta(\cdot, \cdot) : \mathcal{C} \times \mathcal{U} \to \mathbb{R}_+$, which is applicable only for buffer occupancy state $x^b > 0$. Assume that when a transmission is attempted, there is no delay penalty cost, that is $\beta(\cdot, 1) = 0$.
  For channel utilization enhancement, it is assumed that $c(\cdot, u)$ is decreasing and $\beta(\cdot, u)$ is increasing in the channel state, that is the transmission cost is lower and the delay penalty cost is higher for better channel states.

If the transmission of a packet over the channel is attempted, that is action $u = 1$ is selected, a packet will be successfully received (and hence removed from the buffer) with probability given by the function $f : \mathcal{C} \to [0, 1]$. Here, $f(\cdot)$ is a user-defined increasing function, that is, a higher (better) channel state has a higher success probability.

The transition probabilities of the constrained MDP are then given by

$$\begin{aligned}
\mathbb{P}(X_{n+1}|X_n, u = 0) &= \mathbb{P}(X_{n+1}^c|X_n^c)\mathbb{P}(X_{n+1}^y)\mathbf{I}(X_{n+1}^b = X_n^b + X_n^y) \\
\mathbb{P}(X_{n+1}|X_n, u = 1) &= \mathbb{P}(X_{n+1}^c|X_n^c)\mathbb{P}(X_{n+1}^y)f(X_n^c)\mathbf{I}(X_{n+1}^b = X_n^b + X_n^y - 1) \\
&\quad + \mathbb{P}(X_{n+1}^c|X_n^c)\mathbb{P}(X_{n+1}^y)(1 - f(X_n^c))\mathbf{I}(X_{n+1}^b = X_n^b + X_n^y),
\end{aligned}$$

where $\mathbf{I}(\cdot)$ is the indicator function. The corresponding constrained MDP is then

given by (2), (3) with only one constraint, that is, $L = 1$:

$$\lim_{N\to\infty} \sup \frac{1}{N}\mathbb{E}_u\left[\sum_{n=1}^{N} \beta(X_n^c, u_n)\right] \leq \gamma. \tag{53}$$

In the remainder of the section we will outline the steps involved in proving the threshold structure of the optimal policy of the above constrained MDP and describe how the threshold structure can be exploited in the proposed weak derivative based stochastic approximation algorithm.

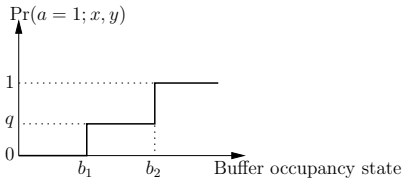

Fig. 1. The constrained average cost optimal policy $\mathbf{u}^*([x^b, x^c, x^y])$ is a randomized mixture between two policies that are deterministic and monotonically increasing in the buffer occupancy state x^b.

The steps involved in proving the structural result for the considered countable state, infinite horizon average cost constrained MDP includes

- Derive a condition for which all constrained policies induce a stable buffer and recurrent Markov chains.
- Use the Lagrange multiplier formulation and prove the existence of an unconstrained optimal policy under the condition for buffer stability.
- Prove the threshold structure of the unconstrained (Lagrangian cost) optimal policies by using the supermodularity concept. The structure of the constrained optimal policy then follows due to a well-known result that relates unconstrained and constrained optimal policies [8; 2].

The condition for buffer stability and recurrence of the Markov chains is as follows, see [22] for proof.

Lemma 2. *Denote* $\min_{x^c\in\mathcal{C}}\{f(x^c)\} = \underline{f}$ *;* $\min_{x^c\in\mathcal{C}} \beta(x^c, 0) = \underline{\beta}$*. If* $\frac{\delta}{\underline{f}} < 1 - \frac{\gamma}{\underline{\beta}}$*, then every policy* $\mathbf{u}$ *satisfying the constraint (53) induces a stable buffer, and a recurrent Markov chain.*

Lagrange formulation and existence of an optimal policy

We convert the constrained MDP to an unconstrained MDP by the Lagrange multiplier method. In particular, for a Lagrange multiplier λ, the instantaneous Lagrangian cost at time n is $c(X_n, u; \lambda) = c(X_n^c, u) + \lambda\beta(X_n^c, u)$. The Lagrangian average cost for a policy $\mathbf{u}$ is then given by

$$J_{x_0}(\mathbf{u}; \lambda) = \lim \sup_{N\to\infty} \frac{1}{N}\mathbb{E}_{\mathbf{u}}\left[\sum_{n=1}^{N} c(X_n, u_n; \lambda)|X_0 = x_0\right], \tag{54}$$

and the corresponding unconstrained MDP is to minimize the above Lagrangian average cost.

The existence and threshold structure of an unconstrained stationary average Lagrangian cost optimal policy are established by viewing the average cost MDP model as a limit of discounted cost MDPs with discount factors approaching 1. In particular, [26; 27] provide the theory for relating average cost optimal policies to discounted cost optimal policies. Define the discounted cost as below

$$J_{x_0}^{\nu}(\mathbf{u}; \lambda) = \lim_{N\to\infty} \mathbb{E}_{\mathbf{u}}\left[\sum_{n=1}^{N} \nu^{n-1} c(X_n, u_n; \lambda)|X_0 = x_0\right], \tag{55}$$

where $0 \leq \nu \leq 1$ is the discount factor. Define the optimal discounted cost by $V^{\nu}(x_0) = \inf_{u\in\mathcal{D}} J_{x_0}^{\nu}(\mathbf{u}; \lambda)$ (for notational convenience we omit the notation of Lagrange multiplier λ in $V^{\nu}(x_0)$).

In [26; 27], the authors proved that the average cost optimal policy exists and inherits the structure of the discounted cost optimal policies under the following conditions:

A1. For each state x and discount factor ν, the optimal discounted cost $V^{\nu}(x)$ is finite.

A2. Assume a reference state 0. There exists a nonnegative N such that $-N \leq h_{\nu}(x) \stackrel{\triangle}{=} V^{\nu}(x) - V^{\nu}(0)$ for all $x \in \mathcal{S}$ and $\nu \in (0, 1)$.

A3. There exists nonnegative M_x, such that $h_{\nu}(x) \leq M_x$ for every $x \in \mathcal{S}$ and ν. For every x there exists an action $u(x)$ such that $\sum_{x'} \mathbb{P}(x'|x, u(x))M_{x'} < \infty$.

Define the reference state by $X^0 = [x^b = 0, x^c = \mathcal{C}_K, x^y = 0]$. In light of Lemma 2 it is clear that the policy of always transmitting whenever the buffer is not empty will induce a stable buffer, and hence finite expected time and cost for first passage to the reference state. As a result, due to Propositions 5(i) and 4(ii) in [27], A1 and A3 hold. Furthermore, as all instantaneous costs are bounded, the following value iteration converges to the optimal discounted cost $V^{\nu}(\cdot)$ for all

discount factor ν

$$V^{\nu}_{n+1}(x) = \min_{u\in\mathcal{U}} Q^{\nu}_{n+1}(x,u) \tag{56}$$

$$Q^{\nu}_{n+1}(x,u) = c(x,u;\lambda) + \nu \sum_{x'\in\mathcal{S}} \mathbb{P}(x'|x,u) V^{\nu}_{n}(x',a). \tag{57}$$

Using the recursion (56)–(57) it can be shown by induction that $V^{\nu}([x^b,x^c,x^y])$ is increasing in x^b and x^y [21], which implies that A2 holds.

Threshold structure of discounted/average cost optimal policies

Due to convergence of (56), (57) for all initial conditions, in order to show that the discounted cost optimal policy is monotonically increasing in the buffer state b it suffices to show $Q^{\infty}([x^b,x^c,x^y],u)$ is submodular in (x^b,u) for all $x^b \geq 1$ for some initial condition [29]. This can be done via mathematical induction as in the theorem below, see [22] for proof.

Theorem 4. *The discounted (Lagrangian) cost optimal policy is a threshold policy of the form*

$$u^*_{\nu}([x^b,x^c,x^y]) = \begin{cases} 0 \text{ if } 0 \leq x^b < b(x^c,x^y) \\ 1 \text{ if } b(x^c,x^y) \leq x^b, \end{cases} \tag{58}$$

where $b(x^c,x^y) : \mathcal{C}\times\mathcal{Y} \to \mathcal{B}$ defines the threshold for the pair of channel state and packet arrival event (x^c,x^y).

Therefore (unconstrained) Lagrangian average cost optimal policy, which inherits the threshold structure of some sequence of discounted cost optimal policies, is of the form (58). Due to Theorem 4.3 in [8], the constrained optimal policy for the constrained MDP is a randomized mixture of two threshold policies:

$$\mathbf{u}^* = q(x^c,x^y)\mathbf{u}^*_1 + (1-q(x^c,x^y))\mathbf{u}^*_2 = \begin{cases} 0 & \text{if } 0 \leq x^b < b_1(x^c,x^y) \\ q(x^c,x^y) & \text{if } b_1(x^c,x^y) < x^b < b_2(x^c,x^y) \\ 1 & \text{if } x^b > b_2(x^c,x^y). \end{cases} \tag{59}$$

Here for each channel state $x^c \in \mathcal{C}$ and packet arrival state $x^y \in \mathcal{Y}$, $q(x^c,x^y) \in [0,1]$ denotes the mixture probability and $\mathbf{u}^*_1, \mathbf{u}^*_2$ are monotone policies in the buffer state x^b of the form (58) with threshold states $b_1(x^c,x^y)$ and $b_2(x^c,x^y)$, respectively. Therefore, the optimal policy $\mathbf{u}$ has a simple threshold structure.

7. Numerical Examples

7.1. *Comparison of Efficiency of Gradient Estimators*

Since efficient gradient estimation of the costs and constraints is an essential ingredient of the adaptive control algorithms proposed in this paper, in this subsection we compare the gradient estimators with other gradient estimators.

Score Function Estimator of Bartlett & Baxter [4; 3]: The Score Function gradient estimator of (19) can be derived using the measure-valued approach of [13]. Let an arbitrary finite-valued random variable Z take value j with probability $p_\theta(j)$. Then

$$\frac{\partial}{\partial\theta_{ia}}F(Z)=\frac{\partial}{\partial\theta_{ia}}\sum_j F(j)\,p_\theta(j)=\sum_j\left(\frac{\partial}{\partial\theta_{ia}}\ln[p_\theta(j)]\right)F(j)p_\theta(j)=\mathbb{E}_\alpha[F(Z)S(\theta_{ia},Z)]$$

where $S(\theta_{ia},Z)\stackrel{\triangle}{=}\frac{\partial}{\partial\theta_{ia}}\ln[p_\theta(Z)]$ is known as the Score Function. Returning to the Markov process $\{Z_n\}$, using $P(Z_{k+1}=(i,a)|Z_k=(j,u))=A_{ji}(u)\theta_{ia}$ yields $S(\theta_{ia},Z_{k+1})=\frac{1}{\theta_{ia}}\mathbf{1}_{\{Z_{k+1}=(i,a)\}}$, which is not uniformly bounded in θ: when one or more components of the control parameter tend to zero (which they do when a policy is pure instead of randomized) the estimator blows up. To overcome this problem a Score Function estimator is used in [4; 3] with the exponential parameterization. When inserted in the formula (19) for the chain rule, the Score Function estimator for the Markov Chain is of the form $\sum_n S(\theta_{ia},Z_n)$ and it is a well known problem that the variance increases with time. Numerous variance reduction techniques have been proposed in the literature [23] including regenerative estimation, finite horizon approximations and more recently [4; 3] propose to use a forgetting factor for the derivative estimator. Their method suffers therefore of a variance/bias trade-off, while our estimation method is consistent and has uniformly bounded variance (in N).

System Parameters: We simulated the following MDP: $S=\{1,2\}$ (2 states), $d(i)+1=3$ (3 actions),

$$A(0)=\begin{pmatrix}0.9&0.1\\0.2&0.8\end{pmatrix},\quad A(1)=\begin{pmatrix}0.3&0.7\\0.6&0.4\end{pmatrix},\quad A(2)=\begin{pmatrix}0.5&0.5\\0.1&0.9\end{pmatrix}.$$

The action probability matrix $(\theta(i,a))$ and cost matrix $(c(i,a))$ were chosen as:

$$(\theta(i,a))=\begin{bmatrix}0.2&0.6&0.2\\0.4&0.4&0.2\end{bmatrix},\quad (c(i,a))=-\begin{bmatrix}50.0&200.0&10.0\\3.0&500.0&0.0\end{bmatrix}$$

Gradient Estimates in Spherical Coordinates: The theoretical values of the gradients are $\nabla_\alpha C_N(\alpha)=\begin{pmatrix}45.05&-55.07\\187.58&-159.91\end{pmatrix}$. The gradient estimates (40) for

batch sizes $N = 100, 1000$ are:

$$\widehat{G}_{100} = \begin{pmatrix} 43.333 \pm 4.791 & -54.191 \pm 2.096 \\ 196.956 \pm 8.951 & -161.200 \pm 4.918 \end{pmatrix}$$
$$\widehat{G}_{1000} = \begin{pmatrix} 44.322 \pm 1.323 & -53.656 \pm 0.712 \\ 189.549 \pm 2.829 & -164.329 \pm 1.231 \end{pmatrix}$$

In the above expression, the numbers following the $\pm$ sign are confidence intervals which were estimated at level 0.05 using the normal approximation with 100 batches.

WD Phantoms versus Score Function Method: As mentioned in Sec.2.4 the closest approach to the algorithms in this paper is that in [4; 3], which uses a Score Function method to estimate the gradients. Here we compare our gradient estimator with the score function gradient estimator of [4; 3]. Since the score function gradient estimator in [4; 3] uses canonical coordinates θ, to make a fair comparison in this example we work with canonical coordinates. A detailed development of the appropriate modifications to the corresponding factors multiplying the realisation perturbations is in [1]. The theoretical values of the generalized gradient (19) for the above MDP are

$$\nabla_\psi [C(\theta(\psi))] = \begin{pmatrix} -9.010 & 18.680 & -9.670 \\ -45.947 & 68.323 & -22.377 \end{pmatrix}. \tag{60}$$

We simulated the WD phantom and score function (SF) estimators for (19) in canonical coordinates, see [1] for implementation details. For batch sizes $N = 100$ and 1000, the WD phantom gradient estimates are

$$\widehat{\nabla C}^{\text{WD}}_{100} = \begin{pmatrix} -7.851 \pm 0.618 & 17.275 \pm 0.664 & -9.425 \pm 0.594 \\ -44.586 \pm 1.661 & 66.751 \pm 1.657 & -22.164 \pm 1.732 \end{pmatrix}$$
$$\widehat{\nabla C}^{\text{WD}}_{1000} = \begin{pmatrix} -8.361 \pm 0.215 & 17.928 \pm 0.240 & -9.566 \pm 0.211 \\ -46.164 \pm 0.468 & 68.969 \pm 0.472 & -22.805 \pm 0.539 \end{pmatrix}.$$

Again the numbers after $\pm$ above, denote the confidence intervals at level 0.05 with 100 batches. Note that for some of the elements of $\widehat{\nabla C}^{\text{WD}}$ above, the confidence interval estimates do not contain the theoretical value (60). The variance of the WD phantom estimator is shown in Table 1, together with CPU time.

$N = 1000$	$\mathsf{Var}[\widehat{\nabla C}^{\text{WD}}_N]$		
$i = 0$	1.180	1.506	1.159
$i = 1$	5.700	5.800	7.565
CPU	2 secs.		

We implemented the score function gradient estimator of [4; 3] with the following parameters: forgetting factor 1 (otherwise the estimates are biased), batch sizes of $N = 1000$ and 10000. In both cases a total number of $10,000$ batches were simulated. The score function gradient estimates are

$$\widehat{\nabla C}^{\text{SF}}_{10000} = \begin{pmatrix} -3.49 \pm 5.83 & 16.91 \pm 7.17 & -13.42 \pm 5.83 \\ -41.20 \pm 14.96 & 53.24 \pm 15.0 & -12.12 \pm 12.24 \end{pmatrix}$$

$$\widehat{\nabla C}^{\text{SF}}_{1000} = \begin{pmatrix} -6.73 \pm 1.84 & 19.67 \pm 2.26 & -12.93 \pm 1.85 \\ -31.49 \pm 4.77 & 46.05 \pm 4.75 & -14.55 \pm 3.88 \end{pmatrix}$$

The variance of the score function gradient estimates are given Table 2.

$N = 1000$	$\mathsf{Var}[\widehat{\nabla C}^{\text{SF}}_N]$		
$i = 0$	89083	135860	89500
$i = 1$	584012	593443	393015
CPU	1374 secs.		

$N = 10000$	$\mathsf{Var}[\widehat{\nabla C}^{\text{SF}}_N]$		
$i = 0$	876523	1310900	880255
$i = 1$	5841196	5906325	3882805
CPU	13492 secs.		

Note that even with substantially larger batch sizes and number of batches (and hence computational time), the variance of the score function estimator is orders of magnitude larger than that of the WD phantom estimator.

7.2. *Stochastic Adaptive Control of time varying Constrained MDP*

We consider adaptive stochastic control of the following time-varying constrained MDP: For time up to 4000, $S = \{0, 1\}, \mathcal{U}_i = \{0, 1, 2\}$, $i \in S$ ($d(0) = d(1) = 2$),

$$A(0) = \begin{pmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{pmatrix}, \quad A(1) = \begin{pmatrix} 0.3 & 0.7 \\ 0.6 & 0.4 \end{pmatrix}, \quad A(2) = \begin{pmatrix} 0.5 & 0.5 \\ 0.1 & 0.9 \end{pmatrix}.$$

The cost matrix $(c(i, a))$, two constraints matrices $(\beta_1(i, a))$, $(\beta_2(i, a))$ are

$$(c(i, a)) = -\begin{bmatrix} 50 & 200 & 10 \\ 3 & 500 & 0 \end{bmatrix}, \quad \beta_1 = \begin{bmatrix} 20 & 100 & -8 \\ -3 & 4 & -10 \end{bmatrix}, \; \beta_2 = \begin{bmatrix} 10 & -20 & 22 \\ -19 & 17 & -15 \end{bmatrix}.$$

The optimal control policy incurs a cost of -111.80 (or equivalently a reward of 111.80) and is randomized with probabilities (18) $\theta^* = \begin{bmatrix} 0 & 0.2 & 0.8 \\ 0 & 0.28 & 0.72 \end{bmatrix}$. For time

between 4000 and 12000 the transition probabilities are

$$A(0) = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}, \qquad A(1) = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}, \qquad A(2) = \begin{bmatrix} 0.5 & 0.5 \\ 0.45 & 0.55 \end{bmatrix}.$$

This has an optimal cost of -44.52 (that is reward of 44.52).

The algorithm was initialized with randomized policy $\theta(0) = \begin{bmatrix} 0.1 & 0.1 & 0.8 \\ 0.0 & 0.2 & 0.8 \end{bmatrix}$. The batch sizes over which the gradients are estimated was chosen as $N = 10$. The parameters used in the primal dual algorithm are $\rho = 100$, $\epsilon = 2 \times 10^{-7}$ (see (43), (44)).

As can be seen from Fig.2, it takes only around 100 batches (1000 time points) for the algorithm to rapidly approaches the optimal policy. The algorithm also quickly responds to the change in optimal policy at batch time 400. The choice of the discounting factor δ in (52) of the primal dual method clearly shows the trade off between bias and tracking ability in Fig.2. For $\delta = 1.0$, the algorithm has fast tracking properties but a large bias. For $\delta = 0.5$ and $\delta = 0.1$ the bias gets smaller but the adaptation rate is slower. Our conference paper [17] and report [1] give additional numerical examples.

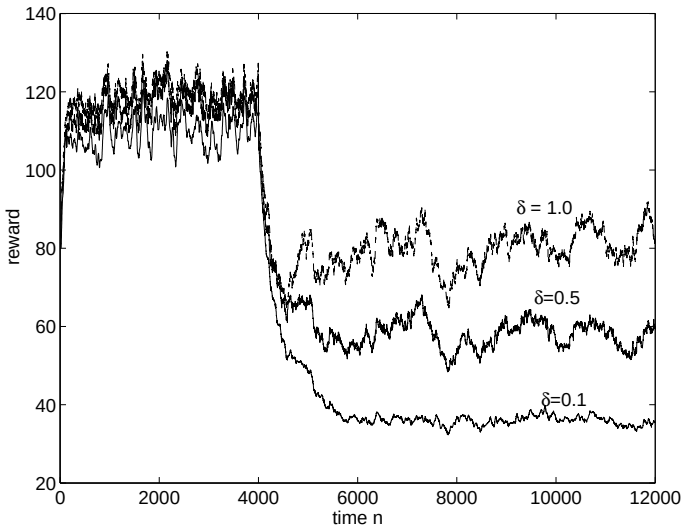

Fig. 2. Primal dual algorithm based stochastic adaptive controller

8. Conclusions

In this paper simulation based gradient algorithms have been presented for adaptively optimizing a constrained average cost finite state MDP. First a parameterization of the randomized control policy using spherical coordinates was presented. Then a novel measure-valued gradient estimator using WD phantoms for short batch sizes was presented. The ensuing algorithm for estimating gradients of the cost and constraints does not require explicit knowledge of the transition probabilities of the MDP or any off-line simulations.

As illustrated in Sec.7.1, the resulting gradient estimator has much smaller variance than score function based gradient estimators. The WD phantom gradient estimator was then used in a stochastic gradient algorithm with fixed step size in order to track time varying MDP with unknown transition probability matrices. Primal dual and multiplier based stochastic gradient algorithms were presented for handling the constraints.

9. Appendix: Proof of Theorem 2

To relate our formula (40) to the realization perturbation formulas in [9] use the following argument. The matrix $P(\alpha)$ in (9) denotes the transition probability of $\{Z_n\}$ using a fixed value of the control parameter α. Fix the indices i, p and consider a perturbation of $\alpha_{i,p}$ into $\alpha_{i,p} + \delta$. The new transition matrix is $P(\alpha + \delta) = P(\alpha) + \delta Q(\alpha) + o(\delta^2)$, for $Q(\alpha) = P'(\alpha)$ the matrix with entries equal to the derivatives of the entries of $P(\alpha)$ with respect to $\alpha_{i,p}$. In [9], it is established that

$$\frac{\partial}{\partial \alpha_{ip}} \mathbb{E}_{\pi(\alpha)}[c(Z)] = \pi(Q(\alpha)g), \tag{61}$$

where π is the stationary state probability of the chain and the vector g has entries:

$$g(i,u) = \lim_{N\to\infty} \mathbb{E}_\alpha \left[\sum_{n=1}^{N} F(Z_n) \mid Z_0 = (i,u) \right] - (N-1)C(\alpha).$$

Let $p = 1, \ldots, d(i) - 1$, then by definition, the matrix Q satisfies:

$$Q_{(j,u),(i',u')} = \frac{\partial}{\partial \alpha_{ip}} P_{(j,u)(I',u')}(\alpha) = p_{j,i'}(u) \frac{\partial}{\partial \alpha_{ip}} \theta_{i',u'}(\alpha)$$
$$= p_{j,i'}(u) \times \begin{cases} 0 & i' \neq i \\ 0 & i' = i, u' < p-1 \\ -2\tan(\alpha_{ip})\theta_{i',p-1} & i' = i, u' = p-1 \\ +2\tan(\alpha_{ip})\theta_{i',u'} & i' = i, u' \geq p \end{cases}$$

where we have used the expression $\theta_{i,u'} = \cos^2(\alpha_{i,u'+1}) \prod_{m=1}^{u'+1} \sin^2(\alpha_{i,u'+1})$.

Using now $\pi P = \pi$, and the expression for Q, (61) becomes:

$$\frac{\partial}{\partial \alpha_{ip}} \mathbb{E}_{\pi(\alpha)}[c(Z)] = -2\tan(\alpha_{ip})\, \pi_{i,u'} \mathsf{E}_i[g(i,p-1) - g(i,Y_p)].$$

As explained in [9; 1], $g(i,p-1) - g(i,Y_p) = \sum_{n=k}^{k+nu(k)} c(Z_n) - \nu(k)C(\alpha)$, with initial condition $Z_k = (i, p-1)$ and $nu(k) = \min\{n > 0 : Z_{k+n} = (i, Y_p)\}$, so the resulting expression is:

$$\frac{\partial}{\partial \alpha_{ip}} \mathbb{E}_{\pi(\alpha)}[c(Z)] = -2\tan(\alpha_{ip})\theta_{i,p-1} \mathsf{E}_i \left[\sum_{n=k}^{k+nu(k)} \mathbb{E}_\alpha[c(Z_n) - nu(k)C(\alpha) \mid Z_k = (i, p-1)] \right] \tag{62}$$

for $p = 1, \ldots, d(i) - 1$. When implementing the above equation on-line, we use the observations of the actual process $\{Z_k\}$ to estimate the various derivatives. The proportion of time that we observe action $p-1$ at state i is, by definition, $\theta_{i,p-1}$, thus the averaging over N observations yields

$$\frac{\partial}{\partial \alpha_{ip}} \mathbb{E}_{\pi(\alpha)}[c(Z)] = -2\tan(\alpha_{ip}) \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \mathsf{E}_i \left[\delta_{i,p}(k) \sum_{n=k}^{k+\nu(k)} \mathbb{E}_\alpha[c(Z_n) - \nu(k)C(\alpha) \right] \tag{63}$$

The result (40) is established noticing that $\hat{\Gamma}_N$ converges a.s. to $C(\alpha)$, because for $\alpha \in \boldsymbol{\alpha}^\mu$ $\mathbb{E}_\alpha[\nu(k)] < \infty$, being a coupling time of a positive recurrent Markov chain. The case $p = d(i)$ consists of a degenerate case and a similar chain rule argument can be shown for the factor $K_{i,d(i)}(\alpha, k)$. A detailed derivation of this formula can be obtained also using measure-valued differentiation [23; 13], and appears in [1].

10. Appendix: Proof of Proposition 1.2

Proof. First, from tightness of $\{\lambda^\epsilon(n)\}$, it follows that the family $\{\bar{\lambda}(\epsilon)\}$ of (deterministic) averages lies in a compact set, thus the accumulation points $\bar{\lambda}$ exist.

From the proof of uniform integrability in Proposition 1.1, and tightness of $\{\lambda^\epsilon(n)\}$, it follows that the sequence $\{(\alpha^\epsilon(n+1) - \alpha^\epsilon(n)/\epsilon)\}$ is uniformly integrable, which implies that $\{\alpha^\epsilon(n)\}$ is tight. Therefore for any sequence $(\alpha^{\epsilon_k}(\cdot), \lambda(\epsilon_k))$ there is at least one (weakly) convergent subsequence with a.s. Lipschitz continuous limit (refer to [19]). For the rest of the proof, until specified, assume that ϵ labels a weakly convergent subsequence (to avoid the ϵ_k cumbersome indexing). We will now identify the limits of such convergent subsequences and show that they all satisfy the same ODE. Also to ease the notation in the proof, call $Y_0(n) = \widehat{\nabla_\alpha C}(n), Y_l(n) = \widehat{\nabla_\alpha B_l}(n)$.

From the definition (24) it follows that:

$$\alpha^\epsilon(t+s) - \alpha^\epsilon(t) = -\epsilon \sum_{n=\lfloor t/\epsilon \rfloor}^{\lfloor (t+s)/\epsilon \rfloor - 1} \left[Y_0(n) + \sum_{l=1}^{L} \lambda_l^\epsilon(n+1)\, Y_l(n) \right].$$

Divide now the interval $(t, t+s]$ into subintervals of small size δ_ϵ containing each a number n_ϵ of updates, as shown in Figure 3.

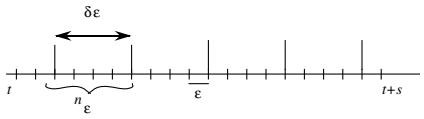

Fig. 3. $\delta_\epsilon = \epsilon n_\epsilon$. Condition: $\delta_\epsilon \to 0, n_\epsilon \to \infty$ as $\epsilon \to 0$.

Using the δ_ϵ grouping of subintervals, one obtains the telescopic sum:

$$\alpha^\epsilon(t+s) - \alpha^\epsilon(t) = - \sum_{l=\lfloor t/\delta_\epsilon \rfloor}^{\lfloor (t+s)/\delta_\epsilon \rfloor - 1} \delta_\epsilon \times \left(\frac{1}{n_\epsilon} \sum_{j=ln_\epsilon}^{(l+1)n_\epsilon - 1} \left[Y_0(j) + \sum_{l=1}^{L} \lambda_l^\epsilon(j+1)\, Y_l^\epsilon(j) \right] \right).$$

We now show that if $\mathfrak{F}^\epsilon(t)$ denotes the σ-algebra generated by the interpolated process up to time t, then:

$$\mathsf{E}_i[\alpha^\epsilon(t+s) - \alpha^\epsilon(t) \mid \mathfrak{F}^\epsilon(t)] \approx \sum_{l=\lfloor t/\delta_\epsilon \rfloor}^{\lfloor (t+s)/\delta_\epsilon \rfloor - 1} \delta_\epsilon \left(h_0[\alpha^\epsilon(ln_\epsilon)] + \sum_{l=1}^{L} (\bar{\lambda}_l(\epsilon) + B_l[\alpha^\epsilon(ln_\epsilon)])\, h_l[\alpha^\epsilon(ln_\epsilon) + \kappa[\alpha^\epsilon(ln_\epsilon)] \right) \quad (64)$$

where the expectation of the absolute error in the end points of the δ_ϵ discretization vanishes as $\epsilon \to 0$. Let $\mathbb{E}_{ln_\epsilon}$ denote the expectation condition-

ing on the information available up to the start of the current small subinterval of size δ_ϵ. Use now conditional expectations to express $\mathbb{E}[Y_l(j) \mid \mathfrak{F}^\epsilon(t)] = \mathbb{E}[\mathbb{E}_{ln_\epsilon}[Y_l(j)] \mid \mathfrak{F}^\epsilon(t)], ln_\epsilon \le j < (l+1)n_\epsilon$ for each term in the telescopic sum $l = 0, \ldots, L$. That is, we use a filter of the terms, focusing on each of the averages within subintervals.

Because one is interested in averages, any version of the process $\{\alpha^\epsilon(n)\}$ can be used to characterize these conditional expectations. In particular, Skorohod representation establishes that there is a process $\tilde{\alpha}^\epsilon(n)$ for each ϵ in the weakly convergent subsequence, such that $\tilde{\alpha}^\epsilon(n)$ has the same distribution as $\alpha^\epsilon(n)$ and it converges with probability 1 to the same a.s. continuous limit $\alpha(t)$ (see [19]). Because $\alpha(t)$ is Lipschitz continuous w.p.1, $\|\alpha(l\delta_\epsilon + \delta_\epsilon) - \alpha(l\delta_\epsilon)\| = \mathcal{O}(\delta_\epsilon)$ and since $\tilde{\alpha}^\epsilon(n)$ converges w.p. 1, it follows that $\sup_{ln_\epsilon \le j < (l+1)n_\epsilon} \|\tilde{\alpha}^\epsilon(j) - \alpha(l\delta_\epsilon)\| \to 0$ a.s. which implies that the underlying distribution of the batch estimators $Y_l(j), j = ln_\epsilon, \ldots, (l+1)n_\epsilon - 1$ converges to that of the fixed-α MDP at the parameter value $\alpha = \alpha(l\delta_\epsilon)$. Using the fact that $\alpha^\epsilon(ln_\epsilon)$ converges in distribution to $\alpha(l\delta_\epsilon)$, it follows that:

$$\mathbb{E}_{ln_\epsilon}\left[\frac{1}{n_\epsilon}\sum_{j=ln_\epsilon}^{(l+1)n_\epsilon-1}\left[Y_0(j) + \sum_{l=1}^{L}\lambda_l^\epsilon(j+1)\, Y_l(j)\right]\right]$$
$$= \mathbb{E}_{ln_\epsilon}\left[\frac{1}{n_\epsilon}\sum_{j=ln_\epsilon}^{(l+1)n_\epsilon-1}\mathbb{E}_\alpha\left[\widehat{\nabla_\alpha C}_0(j) + \sum_{l=1}^{L}(\lambda_l^\epsilon(j) + \hat{B}(j))\,\widehat{\nabla_\alpha B_l}(j)\right]\right].$$

Given the initial state value (with the aggregated information about the living phantoms), and the value of $\lambda_l^\epsilon(j)$, the expectation for the fixed α process of the gradient estimator $\widehat{\nabla_\alpha B_l}(j)$ is independent of $\lambda_l^\epsilon(j)$. As $n_\epsilon \to \infty$, the underlying process $\{Z_n\}$ will have the stationary distribution for the j-th estimation batch, so that:

$$\mathbb{E}_{ln_\epsilon}\left[\frac{1}{n_\epsilon}\sum_{j=ln_\epsilon}^{(l+1)n_\epsilon-1}\left[Y_0(j) + \sum_{l=1}^{L}\lambda_l^\epsilon(j+1)\, Y_l(j)\right]\right]$$
$$= h_0[\alpha^\epsilon(ln_\epsilon)] + \sum_{l=1}^{L}\left(h_l[\alpha^\epsilon(ln_\epsilon)]\,\mathbb{E}_{ln_\epsilon}\left[\frac{1}{n_\epsilon}\sum_{j=ln_\epsilon}^{(l+1)n_\epsilon-1}\lambda_l^\epsilon(j)\right] + \rho\,\mathbb{E}_{\pi(\alpha)}[\hat{B}_l(n)\,\widehat{\nabla_\alpha B_l}(n)]\right)$$
$$\approx h_0[\alpha^\epsilon(ln_\epsilon)] + \sum_{l=1}^{L}(\bar{\lambda}(\epsilon) + \rho B[\alpha^\epsilon(ln_\epsilon)])h_l[\alpha^\epsilon(ln_\epsilon)] + \kappa[\alpha^\epsilon(ln_\epsilon)],$$

which establishes (64). Define now a piecewise constant function (on the δ_ϵ-

subintervals):

$$\mathcal{G}^\epsilon(\alpha^\epsilon(t),\bar{\lambda}(\epsilon)) = h_0[\alpha^\epsilon(ln_\epsilon)]+\sum_{l=1}^{L}(\bar{\lambda}_l(\epsilon)+\rho B[\alpha^\epsilon(ln_\epsilon)])h_l[\alpha^\epsilon(ln_\epsilon)]+\kappa[\alpha^\epsilon(ln_\epsilon)],$$

for $\delta_\epsilon \leq t < (l+1)\delta_\epsilon$, then (64) implies that: $\mathbb{E}[\alpha^\epsilon(t+s) - \alpha^\epsilon(t) \mid \mathfrak{F}^\epsilon(t)] \approx \int_t^{t+s} \mathcal{G}^\epsilon[\alpha^\epsilon(s),\bar{\lambda}(\epsilon)]\,ds$ which implies that the limit process is a martingale with zero quadratic variation. For a detailed presentation of this methodology the reader is referred to [19]. Taking now the limit along the weakly convergent subsequence, $\alpha^\epsilon(t) \to \alpha(t), \bar{\lambda}(\epsilon) \to \bar{\lambda}$ establishes the limiting ODE for this subsequence.

□

References

1. F. Vazquez Abad and V. Krishnamurthy. Self learning control of constrained Markov decision processes– a valued gradient approach. Technical Report G-2003-51, GERAD-HEC Montreal, http://www.gerad.ca/fichiers/cahiers/G-2003-51.pdf, August 2003.
2. E. Altman. *Constrained Markov Decision Processes*. Chapman and Hall, London, 1999.
3. P. Bartlett and J. Baxter. Estimation and approximation bounds for gradient-based reinforcement learning. *J. Comput. Syst. Sci.*, 64(1):133–150, 2002.
4. J. Baxter and P. Bartlett. Direct gradient-based reinforcement learning: I. Gradient estimation algorithms. Technical report, Computer Sciences Laboratory, Australian National University, http://discus.anu.edu.au/ml/index.html, 1999.
5. A. Benveniste, M. Metivier, and P. Priouret. *Adaptive Algorithms and Stochastic Approximations*, volume 22 of *Applications of Mathematics*. Springer-Verlag, 1990.
6. D. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA., 2000.
7. D. Bertsekas and J.N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA., 1996.
8. F. J. Beutler and K. W. Ross. Optimal Policies for Controlled Markov Chains with a Constraint. *Journal of Mathematical Analysis and Applications*, 112:236–252, 1985.
9. X. R. Cao. The relations amongst potentials, perturbation analysis and Markov decision processes. *J DEDS*, 8:71–87, 1998.
10. C. Derman, G.J. Lieberman, and S.M. Ross. Optimal system allocations with penalty cost. *Management Science*, 23(4):399–403, December 1976.
11. O.N. Gharehshiran and V. Krishnamurthy. Coalition Formation for Bearings-Only Localization in Sensor Networks – A Cooperative Game Approach. *IEEE Trans. Signal Proc.*, 58(8):4322–4338, 2010.
12. B. Heidergott and X-R Cao. A note on the relation between weak derivatives and perturbation realization. *IEEE Trans. Auto. Control*, 47(7):1112–1115, 2002.
13. B. Heidergott and F.J. Vázquez-Abad. Measure valued differentiation for markov chains. *Journal of Optimization, Theory and Applications*, 2007. (to appear).

14. V. Krishnamurthy. Bayesian sequential detection with phase-distributed change time and nonlinear penalty – a lattice programming pomdp approach. *IEEE Trans. Inform. Theory*, 57(3), Oct. 2011. http://arxiv.org/abs/1011.5298.
15. V. Krishnamurthy and D. Djonin. Structured threshold policies for dynamic sensor scheduling–a partially observed Markov decision process approach. *IEEE Trans. Signal Proc.*, 55(10):4938–4957, Oct. 2007.
16. V. Krishnamurthy and D.V. Djonin. Optimal threshold policies for multivariate POMDPs in radar resource management. *IEEE Transactions on Signal Processing*, 57(10), 2009.
17. V. Krishnamurthy and F. Vazquez Abad K. Martin. Implementation of gradient estimation to a constrained Markov decision problem. In *IEEE Conference on Decision and Control*, Maui, Hawaii, 2003.
18. H.J. Kushner and D.S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, 1978.
19. H.J. Kushner and G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, New York, 1997.
20. D.G. Luenberger. *Linear and Nonlinear Programming*. Addison Wesley, Second edition, 1984.
21. M. H. Ngo and V. Krishnamurthy. On Optimality of Monotone Channel-aware Transmission Policies: A Constrained Markov Decision Process approach. In *Proceedings of ICASSP'07*, Honolulu, HI, April 2007.
22. M.H. Ngo and V. Krishnamurthy. Optimality of threshold policies for transmission scheduling in correlated fading channels. *Communications, IEEE Transactions on*, 57(8):2474–2483, 2009.
23. G. Pflug. *Optimization of Stochastic Models: The Interface between Simulation and Optimization*. Kluwer Academic Publishers, 1996.
24. M. Puterman. *Markov Decision Processes*. John Wiley, 1994.
25. K.W. Ross and R. Varadarajan. Markov decision processes with sample path constraints: The communication case. *Operations Research*, 37(5):780–790, Sept-Oct 1989.
26. S. Ross. *Introduction to Stochastic Dynamic Programming*. Academic Press, San Diego, California., 1983.
27. L. I. Sennott. Average Cost Optimal Stationary Policies in Infinite State Markov Decision Processes with Unbounded Costs. *Operations Research*, 37(4):626–633, July-August 1989.
28. V. Solo and X. Kong. *Adaptive Signal Processing Algorithms – Stability and Performance*. Prentice Hall, N.J., 1995.
29. D.M. Topkis. *Supermodularity and Complementarity*. Princeton University Press, 1998.
30. F. Vazquez-Abad. Strong points of weak convergence: a study using RPA gradient estimation for automatic learning. *Automatica*, 35(7):1255–1274, 1999.
31. H. S. Wang and N. Moayeri. Finite-state Markov channel - A useful model for radio communications channels. *IEEE Trans. Vehicular Tech.*, 44(1):163–171, 1995.
32. G. Yin, V. Krishnamurthy, and C. Ion. Regime switching stochastic approximation algorithms with application to adaptive discrete stochastic optimization. *SIAM Journal on Optimization*, 14(4):117–1215, 2004.

33. F. Yu and V. Krishnamurthy. Optimal joint session admission control in integrated wlan and cdma cellular network. *IEEE Transactions Mobile Computing*, 6(1):126–139, Jan. 2007.
34. A. Zhang and S. A. Kassam. Finite-state Markov model for Rayleigh fading channels. *IEEE Trans. Commun.*, 47:1688–1692, November 1999.